



**NAVAL
POSTGRADUATE
SCHOOL**

MONTEREY, CALIFORNIA

THESIS

**GENETIC ALGORITHMS IN THE BATTLESPACE
TERRAIN REASONING AND AWARENESS—BATTLE
COMMAND (BTRA-BC) BATTLE ENGINE**

by

Kenneth J. Maroon

September 2009

Thesis Co-Advisors: Curtis Blais
 Jeff Appleget

**This thesis was done at the MOVES Institute
Approved for public release; distribution is unlimited**

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2009	3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE Genetic Algorithms in the Battlespace Terrain Reasoning and Awareness-Battle Command (BTRA-BC) Battle Engine		5. FUNDING NUMBERS	
6. AUTHOR(S) Kenneth J. Maroon		8. PERFORMING ORGANIZATION REPORT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.	
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) The U.S. Army Corps of Engineers Research and Development Center's (ERDC) Battlespace Terrain Reasoning and Awareness-Battle Command (BTRA-BC) Battle Engine (BBE) is a tool that enables command staffs to semi-automate the creation and evaluation of potential courses of action (COA) for use in military planning. The BBE uses data generated during the initial steps of the Military Decision Making Process (MDMP) and a genetic algorithm to produce and evaluate a population of COAs. This thesis provides a basic background of both the MDMP and genetic algorithms. It describes the features of the BBE and the parameters that control its genetic algorithm. The thesis describes an experiment to test the genetic algorithm parameter effects on the BBE search results. The results of this experiment and research are used to provide recommendations to improve the performance and functionality of the BBE.			
14. SUBJECT TERMS Genetic Algorithms, Military Decision Making Process, Mission Planning, Course of Action, Wargaming, Battle Simulation			15. NUMBER OF PAGES 119
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**GENETIC ALGORITHMS IN THE BATTLESPACE TERRAIN REASONING AND
AWARENESS-BATTLE COMMAND (BTRA-BC) BATTLE ENGINE**

Kenneth J. Maroon
Lieutenant, United States Navy
B.S., United States Naval Academy, 2001

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN
MODELING, VIRTUAL ENVIRONMENTS AND SIMULATION (MOVES)**

from the

**NAVAL POSTGRADUATE SCHOOL
September 2009**

Author: Kenneth J. Maroon

Approved by: Curtis Blais
Co-Advisor

Jeff Appleget
Co-Advisor

Mathias Kölsch
Chairman, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The U.S. Army Corps of Engineers Research and Development Center's (ERDC) Battlespace Terrain Reasoning and Awareness-Battle Command (BTRA-BC) Battle Engine (BBE) is a tool that enables command staffs to semi-automate the creation and evaluation of potential courses of action (COA) for use in military planning. The BBE uses data generated during the initial steps of the Military Decision Making Process (MDMP) and a genetic algorithm to produce and evaluate a population of COAs. This thesis provides a basic background of both the MDMP and genetic algorithms. It describes the features of the BBE and the parameters that control its genetic algorithm. The thesis describes an experiment to test the genetic algorithm parameter effects on the BBE search results. The results of this experiment and research are used to provide recommendations to improve the performance and functionality of the BBE.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	THE MILITARY DECISION-MAKING PROCESS	1
B.	FASTER COA CREATION	2
C.	GENETIC ALGORITHMS	3
D.	REASEARCH QUESTIONS	4
E.	SCOPE	4
F.	ORGANIZATION OF THESIS	5
II.	BACKGROUND	7
A.	INTRODUCTION	7
B.	THE MILITARY DECISION-MAKING PROCESS	7
1.	Receipt of Mission	7
2.	Mission Analysis	8
3.	COA Development	9
4.	COA Analysis (War Game)	11
a.	Belt War Gaming	11
b.	Avenue-in-depth War Gaming	12
c.	Box War Gaming	12
5.	COA Comparison	13
6.	COA Approval	14
7.	Orders Production	14
C.	GENETIC ALGORITHMS	16
1.	Advantages of Genetic Algorithms	17
2.	The Basic Genetic Algorithm	19
3.	Population Controls	20
4.	Parent Selection	21
a.	Tournament Selection	21
b.	Roulette Wheel Selection	22
5.	Reproduction and Mutation	23
a.	One-point Crossover	24
b.	Schemata	24
c.	Mutation	26
d.	Two-point Crossover	27
6.	Convergence	28
7.	Niches	29
D.	SUMMARY	33
III.	THE BTRA-BC BATTLE ENGINE	35
A.	INTRODUCTION	35
B.	MISSION ANALYSIS AND INITIAL INPUTS	35
1.	Terrain	36
2.	Order of Battle	40
3.	Postures and Morale	44
C.	COA CREATION	45

1.	Offensive COA Variables	47
2.	Defensive COA Variables	51
D.	WARGAMING	54
1.	Desired End State	54
2.	Battle Visualization	55
3.	COA Evaluations	56
E.	THE BBE GENETIC ALGORITHM	57
1.	Results and Analysis	57
2.	Genetic Algorithm Parameters	61
F.	SUMMARY	63
IV.	METHODS	65
A.	INTRODUCTION	65
B.	THE SCENARIO	65
C.	PARAMETERS TESTED	67
1.	Analysis of Impact of Selection Technique	67
2.	Analysis of Impact of Replacement Policies ...	68
3.	Analysis of the Crowding Model	68
D.	DATA COLLECTION	69
E.	SUMMARY	69
V.	RESULTS AND ANALYSIS	71
A.	INTRODUCTION	71
B.	GENERAL STATISTICS	71
C.	SELECTION TECHNIQUE	72
D.	CROSSOVER	72
E.	REPLACEMENT POLICY	73
F.	THE CROWDING MODEL	74
G.	SUMMARY	75
VI.	CONCLUSIONS AND RECOMMENDATIONS	77
A.	INTRODUCTION	77
B.	CONCLUSIONS	77
C.	IMPROVEMENTS TO THE BBE	78
D.	FUTURE WORK	80
E.	SUMMARY	81
APPENDIX A.	THE GENETIC ALGORITHM TEST SCENARIO—ORDERS OF BATTLE AND ECOAS	83
A.	INTRODUCTION	83
B.	ORDERS OF BATTLE	83
1.	Friendly Order of Battle	83
2.	Enemy Order of Battle	85
C.	ECOAS	88
APPENDIX B.	BBE RUN RESULTS	93
LIST OF REFERENCES	97
INITIAL DISTRIBUTION LIST	99

LIST OF FIGURES

Figure 1.	Wargaming using the yellow sticky drill (From Schlachach, 2008, p. 8).....	2
Figure 2.	The Military Decision-Making Process (From U.S. Army, 1997, pp. 5-2).....	8
Figure 3.	The steps in mission analysis (From U.S. Army, 1997, pp. 5-5).....	9
Figure 4.	An example of COA statement and Sketch (From U.S. Army, 1997, pp. 5-15).....	10
Figure 5.	Belt war gaming (From U.S. Army, 1997, pp. 5-18).....	11
Figure 6.	Avenue-in-depth war gaming (From U.S. Army, 1997, pp. 5-19).....	12
Figure 7.	Box war gaming (From U.S. Army, 1997, pp. 5-19).....	13
Figure 8.	Example decision matrix showing two ways to make comparisons (From U.S. Army, 1997, pp. 5-26).....	14
Figure 9.	Staff inputs and outputs (From U.S. Army, 1997, p. 4).....	15
Figure 10.	Hill climbing can easily find the peak of the first surface, but it may become caught in the local maximum on the second (From Goldberg, 1989, pp. 3-4).....	18
Figure 11.	A roulette wheel constructed from the sample population in Table 1 (From Goldberg, 1989, p. 11).....	23
Figure 12.	Plots for a value (x) over a multi-peak function, $f(x)$. The potential solution populations at generation 100 for two different genetic algorithms, both searching for the value of x that maximizes $f(x)$ are marked as points on the curve and circled. The first genetic algorithm includes sharing in its fitness function, while the second does not (After Goldberg, 1989, p. 194).....	31
Figure 13.	The BBE Mission Building menu.....	36
Figure 14.	The MCOO loader in the Mission Building menu.....	37
Figure 15.	An example of a Braswell Index. The lines "bisect the Mobility Corridors between the obstacles" (From Schlachach, 2008, p. 7).....	38
Figure 16.	BBE game board with several V-Lanes highlighted. The mission start point and objective have also been annotated.....	39
Figure 17.	BBE game board with a LDT circled. Note how it intersects the V-Lanes seen in Figure 16.....	40

Figure 18.	The B-WACT is used to build units for use in the BBE.....	41
Figure 19.	The Weapons and Units Types tab. In this menu, mission planners can load B-WACT weapons lists and build units.....	42
Figure 20.	The Enemy OB tab. The base power of the subordinate command is circled in red.....	43
Figure 21.	The Friendly OB tab. The subordinate command tabs are circled in red.....	43
Figure 22.	Controls for force postures, morale and superiorities.....	44
Figure 23.	The FCOA creation menu with offensive COA variables.....	45
Figure 24.	Graphical visualization of the FCOA displayed in Figure 23. Since this is an offensive COA, units are assigned V-Lanes to define their movement. The exclamation points indicate priority for GS. The values underneath the unit names display the units' percentage of general support, the units' core combat power and the combat power of any subordinate forces Task Organized (TO) to the unit. Each unit's total combat power is displayed in bold.....	47
Figure 25.	The ECOA creation menu with defensive COA variables.....	52
Figure 26.	Graphical visualization of the ECOA displayed in Figure 25. In this defensive COA, units are positioned at LDT-5. Each unit is given a range of V-Lanes that they are responsible for blocking. D Company is set to act as reserves for B Company.....	53
Figure 27.	The Wargame menu.....	54
Figure 28.	The Desired End State menu.....	55
Figure 29.	The Battle visualization screen. Controls at the bottom of the screen allow the user to advance the simulation. Unit information and MC data are displayed to the right of the MCOO.....	56
Figure 30.	An FCOA evaluation matrix.....	57
Figure 31.	The BBE Search Results menu.....	58
Figure 32.	The Cream of the Crop display.....	59
Figure 33.	The Risk Analysis window.....	60
Figure 34.	The Evaluation Criteria Analysis window.....	60
Figure 35.	The Pareto Analysis window.....	61
Figure 36.	The Genetic Algorithm Search Parameters menu. Currently, only items in the on the left side are implemented.....	62

Figure 37.	Battle variables for the test scenario.....	66
Figure 38.	The COA evaluation criteria. Note "sub-MC" criteria are for the mobility corridors at the end of the various V-Lanes.....	67
Figure 39.	Convergence plots from three search runs. The first is from a full Darwin search, the second a half Darwin, and the third a full parricide. In each plot, the black represents the best FCOA score, grey is the average score, and yellow is the worst score.....	74
Figure 40.	General Support Units. The first column indicates if a unit is Task Organizable. The fourth column shows a unit's Base Combat Power (BCP).....	83
Figure 41.	1 st Combined Arms Battalion	84
Figure 42.	2 nd Combined Arms Battalion	84
Figure 43.	Reconnaissance, Surveillance, and Target Acquisition (RTSA) Squadron.....	85
Figure 44.	General Support Units.....	85
Figure 45.	Alpha Company was composed of BMP-2 infantry fighting vehicles.....	86
Figure 46.	Bravo Company.....	86
Figure 47.	Charlie Company.....	87
Figure 48.	Delta Company was composed of T-80U tank platoons.....	87
Figure 49.	Balanced defense with Tank Company in reserve....	88
Figure 50.	Balanced defense with tank platoons integrated into Alpha, Delta, and Charlie Companies.....	89
Figure 51.	Forward defense with tank platoons kept in reserve.....	90
Figure 52.	Strong right defense with tank platoon supporting Alpha Company.....	91
Figure 53.	Strong left defense with Delta Company in reserve of Charlie Company.....	92

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	A sample population of four strings, their fitness scores, and percentage of the total fitness sum (From Goldberg, 1989, p. 11).....	22
Table 2.	The configuration of parameters for crossover, breeding probability and mutation used to test selection techniques.....	68
Table 3.	Comparisons of selection technique performance...	72
Table 4.	Comparisons for crossover technique.....	73
Table 5.	Comparisons of replacement polices.....	73

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AA	Avenue of Approach
AO	Area of Operations
BBE	BTRA-BC Battle Engine
BN	Battalion
BTRA-BC	Battlespace Terrain Reasoning and Awareness- Battle Command
B-WACT	BBE Weapons Assessment and Calculation Tool
C2	Command and Control
COA	Course of Action
CCIR	Commander's Critical Information Requirement
DES	Desired End State
ECOA	Enemy Course of Action
FCOA	Friendly Course of Action
F&S	Follow on Support
GUI	Graphical User Interface
GS	General Support
IPB	Intelligence Preparation of the Battlefield
ISR	Intelligence, Surveillance, and Reconnaissance
JPME	Joint Professional Military Education
LDT	Line of Defensible Terrain
MC	Mobility Corridor
MCOO	Modified Combined Obstacle Overlay
MDMP	Military Decision-Making Process

METT-T	Mission, Enemy, Terrain, Troops and Time
OB	Order of Battle
OBJ	Objective
QJMA	Quantitative Judgment Method of Analysis
RTSA	Reconnaissance, Surveillance, and Target Acquisition
SEED	Simulation Experiments and Efficient Designs
SQD	Squad
TO	Task Organizable
V-Lanes	Virtual Lanes

ACKNOWLEDGMENTS

I would like to start by thanking Curtis Blais for giving me the opportunity to work with the BBE and for his continued guidance throughout the writing process. I would also like to thank Jeff Appleget for assisting me on an extremely short notice. Jerry Schlabach was the key source for information on the BBE and genetic algorithms, and without his expertise in both areas, I do not know if this thesis would have been possible.

I appreciate the hard work and dedication of the entire MOVES faculty and thank them for their professionalism. To my fellow classmates, thank you for all the long hours we spent together working on projects, studying for exams, and joking around. Thank you to my parents, William and Jane Maroon, for giving me a love of learning and pushing me work hard in academics.

Finally, I would like to thank my wife, Aisha. Her love and support over the past two years have made even the longest lectures and toughest assignments seem a little bit more bearable. I could not have had a better partner with whom to enjoy my time here in Monterey. Most importantly, she was the only person who could get me to put away my surfboard and start writing.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. THE MILITARY DECISION-MAKING PROCESS

According to the United States Army Field Manual, the military decision-making process (MDMP) is a "proven analytical process," that can help "the commander and his staff examine a battlefield situation and reach logical decisions" (U.S. Army, 1997, pp. 5-1). While there are several advantages to using the MDMP, its critical weakness is that it is a time-consuming process. A central part of the MDMP is the development, analysis, and comparison of Friendly and Enemy Courses of Action (FCOAs and ECOAs).

At the lowest level, the evaluation of COAs can be accomplished by moving annotated post-it notes across a map of the operational area, also called a "yellow sticky drill" (Schlabach, 2008, p. 8). These post-it notes (or "yellow stickies") represent tactical units and allow planners to see how particular FCOAs and ECOAs fare against each other. As friendly and enemy units are moved across the map, approximations about unit strength and ability are used to resolve engagements and calculate losses. Figure 1 shows a group of planners engaged in a yellow sticky drill.



Figure 1. Wargaming using the yellow sticky drill (From Schlabach, 2008, p. 8)

Ultimately, the number of COAs that could be tested depended on the amount time available for wargaming. The need to make timely decisions may rush mission planners and not allow for an adequate investigation of all possible options.

B. FASTER COA CREATION

The U.S. Army Corps of Engineers Research and Development Center (ERDC) is currently working on a software tool to semi-automate the process of COA creation and evaluation. The Battlespace Terrain Reasoning and Awareness-Battle Command (BTRA-BC) Battle Engine (BBE) is a tool that enables command staff to quickly conduct military planning and evaluation of battle plans. The BBE gives its users the ability to construct tactical units with defined attack, defense, and other attributes. It also incorporates terrain data to provide a map of the operational area with distinct mobility corridors. These building blocks enable planners to quickly construct COAs and conduct a virtual yellow sticky drill. Additionally,

the BBE abstracts COA characteristics, such as unit formations, movements, and responsibilities to a binary string. The COAs can then be evaluated and scored according to the user's desired criteria. Finally, the BBE also has the ability to apply a genetic algorithm to the COA binary string to produce thousands of possible alternative COAs and their respective evaluations.

C. GENETIC ALGORITHMS

Developed by John Holland in the early 1970s, genetic algorithms reflected his belief that features of natural evolution could be used by a computer algorithm to solve difficult problems (Davis, 1991, p. 3). Genetic algorithms rely on a pool of possible solutions to a given problem. These solutions are encoded as binary strings and evaluated. Then a portion of the possible solutions are selected for reproduction, with higher ranking solutions having a greater probability of selection. Reproducing solution strings are paired as parents and recombine with each other to create new child solution strings. Additionally, some strings are mutated, to reflect biological evolution, and provide more diverse solutions. The new strings are evaluated and added to the pool of possible solutions to create the next generation of strings. Since better performing solution strings have a greater probability to be selected for reproduction, each generation should contain increasingly higher performing solutions.

Holland explained the power of genetic algorithms in his schema theorem. He identified patterns in the solution strings called schema. These schemata represented partial

pieces of a total solution. Through the breeding process, more successful schema are brought together to make improved solution strings. This process allows genetic algorithms to evaluate and combine multiple schemata simultaneously.

D. RESEARCH QUESTIONS

The BBE's genetic algorithm allows it to essentially conduct multiple yellow sticky drills for each generation of COAs that it produces. While this is an important feature of the software, it is important to understand how the genetic algorithm behaves and what it is actually producing for the user. Additionally, the BBE gives its users several advanced controls over the genetic algorithm. These capabilities raise the following areas of research:

1. Does the BBE's genetic algorithm actually provide a significant number of useful alternative FCOAs? Can the user separate tactically desirable FCOAs from outcomes that are not logical?
2. How do changes to the genetic algorithm parameters affect the FCOAs that are produced? How do parameters affect the diversity and convergence of the BBE results?
3. Does the BBE provide adequate background on the genetic algorithm and its settings to allow users to understand what they are actually changing? Should the BBE explain the effects, benefits, and limitations of the genetic algorithm settings?

E. SCOPE

The main goal of this thesis is to determine what effects the genetic algorithm parameters have, and to suggest possible improvements to the program regarding the genetic algorithm. This thesis will examine how the

genetic algorithm parameters affect output and the diversity of the suggested FCOAs. The thesis will provide background information on the MDMP and genetic algorithms that are applicable to the BBE. Since any discussion of a particular piece of the BBE will require a basic understanding the complete program, a simple background of the BBE will be provided.

F. ORGANIZATION OF THESIS

This thesis is organized into the following chapters.

Chapter I—Introduction. This chapter gives an abstract description of the MDMP, the BBE, and genetic algorithms. It also identifies the research questions and scope of the thesis.

Chapter II—Background. This chapter has two main focuses, the MDMP and genetic algorithms. The chapter provides basic understanding of both and assists in understanding the BBE functions and settings.

Chapter III—The BBE. This chapter outlines some of the features of the BBE, particularly the parts of the program that involve the genetic algorithm.

Chapter IV—Methods. This chapter describes the evaluation of the BBE's genetic algorithm and its various settings.

Chapter V—Results and Analysis. This chapter outlines the results of the BBE runs analysis of the findings.

Chapter VI—Conclusions and Recommendations. This chapter states what conclusions can be drawn from the evaluation and suggests improvements to the BBE. It also includes the future work section.

II. BACKGROUND

A. INTRODUCTION

Before discussing the BBE, it is important to provide some background information on both the MDMP and genetic algorithms. This chapter gives an outline of the steps in the MDMP and illustrates the time commitment involved. This chapter also provides a basic explanation of genetic algorithms and the different techniques involved in their implementation.

B. THE MILITARY DECISION-MAKING PROCESS

The U.S. Army Field Manual 101 Chapter 5¹ outlines the Military Decision-Making Process (MDMP). According to the field manual, there are seven steps in the process: Receipt of Mission, Mission Analysis, COA Development, COA Analysis, COA Comparison, COA Approval, and Orders Production. Figure 2 illustrates the flow of the MDMP as well as the Commander's responsibilities at each step.

1. Receipt of Mission

The MDMP starts with the receipt of the mission. Critical products of this step include the commander's initial guidance and a warning order. These products outline the time allocation, personnel, and operations necessary to continue the MDMP.

¹ This publication has been superseded by Field Manual 5-0 Army Planning and Orders Production published in January 2005. The steps in the MDMP process are the same and for background information purposes there are no significant changes.

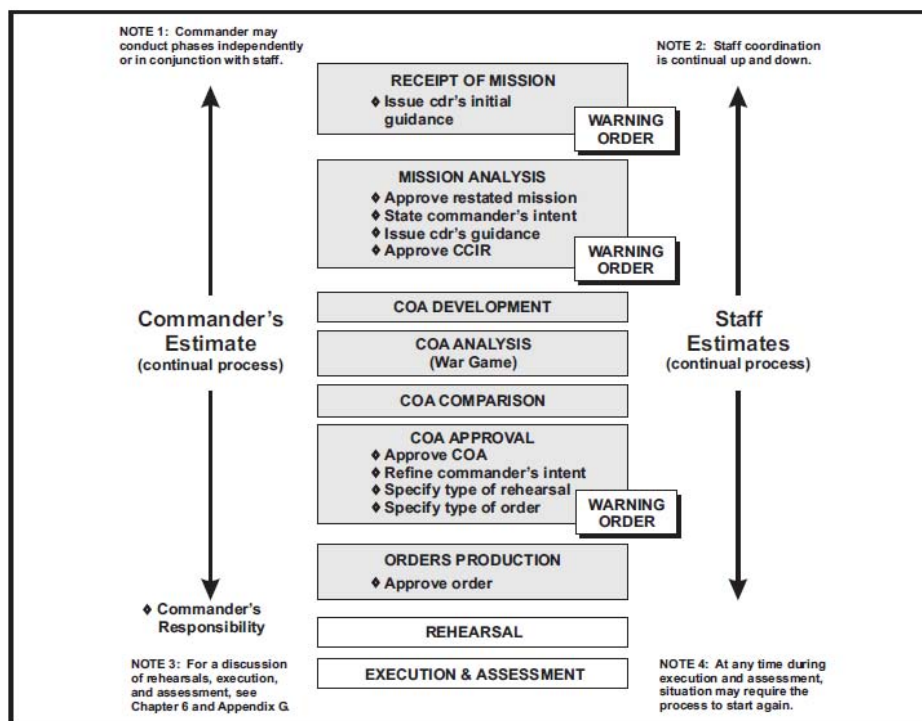


Figure 2. The Military Decision-Making Process (From U.S. Army, 1997, pp. 5-2)

2. Mission Analysis

The second step in the process, mission analysis, is important because "it allows the commander to begin his battlefield visualization" (U.S. Army, 1997, 5-5). The field manual outlines 17 steps for mission analysis, illustrated in Figure 3. These steps do not have to be completed in order. A critical step in mission analysis is to conduct the Intelligence Preparation of the Battlefield (IPB) process. The IPB defines the operational environment, analyzes threats, and identifies possible ECOAs (U. S. Army, pp. 5-6). IPB products such as the enemy order of battle and the modified combined obstacle overlay (MCOO) "are updated and used throughout [the MDMP process]" (U.S. Army, 1997, pp. 5-6). The warning order

containing a summary of information gathered in the previous steps is the final product of mission analysis.

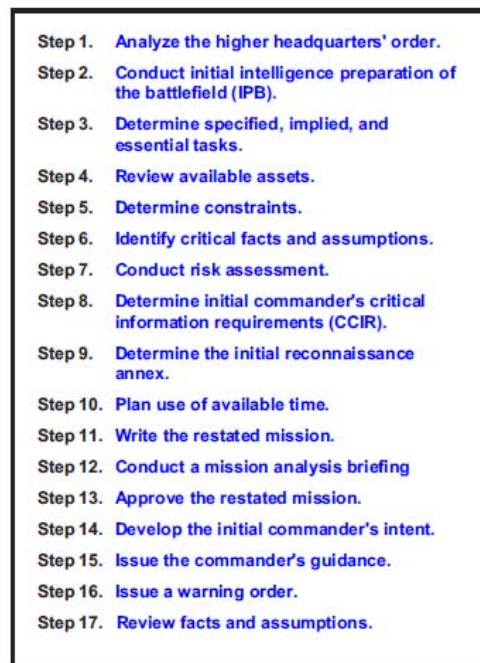
- 
- Step 1. Analyze the higher headquarters' order.
 - Step 2. Conduct initial intelligence preparation of the battlefield (IPB).
 - Step 3. Determine specified, implied, and essential tasks.
 - Step 4. Review available assets.
 - Step 5. Determine constraints.
 - Step 6. Identify critical facts and assumptions.
 - Step 7. Conduct risk assessment.
 - Step 8. Determine initial commander's critical information requirements (CCIR).
 - Step 9. Determine the initial reconnaissance annex.
 - Step 10. Plan use of available time.
 - Step 11. Write the restated mission.
 - Step 12. Conduct a mission analysis briefing
 - Step 13. Approve the restated mission.
 - Step 14. Develop the initial commander's intent.
 - Step 15. Issue the commander's guidance.
 - Step 16. Issue a warning order.
 - Step 17. Review facts and assumptions.

Figure 3. The steps in mission analysis (From U.S. Army, 1997, pp. 5-5)

3. COA Development

In this step, the staff begins to develop plans that are able to meet mission objectives as well as remain aligned with the commander's intent and guidance outlined in the warning order. The COA is expected to meet five basic criteria. The COA must be suitable for the mission. It should be feasible with available resources. It has to be acceptable in terms of risk verse reward. The COA must be distinguishable from other proposed plans of action. Finally, the COA must be a complete statement of the mission and objectives (U.S. Army, 1997, pp. 5-11). It is important note that this step is not complete with the creation of one COA, but instead, generates multiple

Figure 4. An example of COA statement and Sketch (From U.S. Army, 1997, pp. 5-15)

4. COA Analysis (War Game)

The analysis or "war gaming" is the process through which a COA is tested, modified, and improved (U.S. Army, 1997, pp. 5-16). In this step, each COA is enacted on paper against likely ECOAs and evaluated. The criteria for the evaluation are based on the principles of war, doctrinal fundamentals, the commander's intent, and the level of residual risk (U.S. Army, 1997, pp. 5-18). The field manual describes three recommended techniques for war gaming.

a. Belt War Gaming

This style of war gaming divides the area of operations (AO) into a series of distinct sections or belts that run the full width of the battlespace. The belt technique is useful for phased operations, situations facing an enemy arrayed in echelons, or terrain that can be easily divided (U.S. Army, 1997, pp. 5-18). Events in each belt are analyzed in order and any particular belt can contain more than one critical event (U.S. Army, 1997, pp. 5-18). An example of belt war gaming is seen in Figure 5.

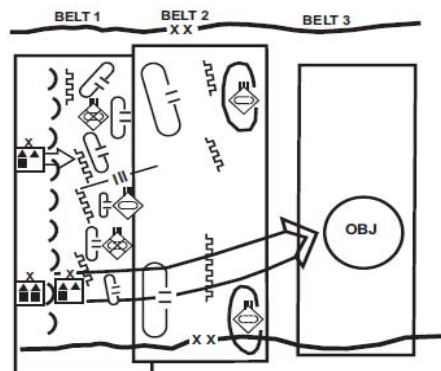


Figure 5. Belt war gaming (From U.S. Army, 1997, pp. 5-18)

b. Avenue-in-depth War Gaming

The avenue-in-Depth technique isolates each avenue of approach in a particular COA and examines it individually. This style is particularly useful in offensive battles or when terrain forces movements in to set channels (U.S. Army, 1997, pp. 5-19). An example is provided in Figure 6.

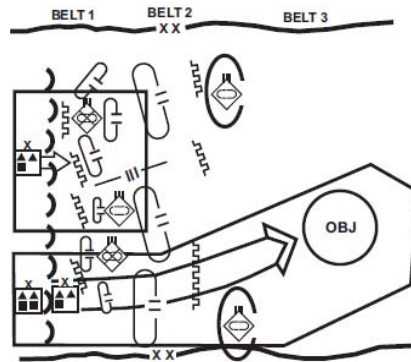


Figure 6. Avenue-in-depth war gaming (From U.S. Army, 1997, pp. 5-19)

c. Box War Gaming

Box war gaming analyzes only important areas of the AO, such as enemy engagement zones or locations of critical objectives. This style is best used in limited time and narrows the focus efforts on essential tasks (U.S. Army, 1997, pp. 5-19). An example of box war gaming is shown in Figure 7.

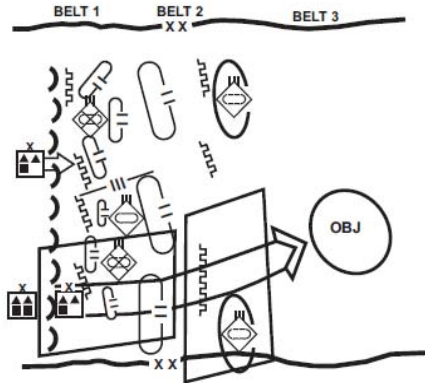


Figure 7. Box war gaming (From U.S. Army, 1997, pp. 5-19)

Once a technique has been selected the war game "follows an action-reaction-counteraction cycle" (U.S. Army, 1997, pp. 5-22). Offensive units initiate an action that the defensive unit responds to with a reaction. The offensive unit can then answer the reaction with a counter action. This cycle continues until completion of the COA or it becomes apparent that the COA is not suitable. War game results are recorded and the acceptable COAs are refined to include updated task organizations and missions.

5. COA Comparison

The main product during this step is the COA decision matrix. The decision matrix uses results from the war gaming exercises to highlight advantages and disadvantages of the COAs that were found to be suitable. There is no standard decision matrix, and each command may have a different style. A decision matrix can have quantitative or subjective information displayed. The only requirement is that it satisfies the commander and it provides a quick, consistent, and accurate comparison of the available COAs. Figure 8 provides an example of a decision matrix showing both subjective analysis and comparison over categories.

Course of Action	Advantages	Disadvantages
COA 1	Main attack avoids major terrain obstacles. Adequate maneuver room for main attack and reserve.	Main attack faces stronger resistance at beginning.
COA 2	Main attack gains good observation early. Supporting attack provides flank protection to main attack.	Initially, reserve may have to be employed in zone of supporting attack. Needs detailed and rehearsed procedural and positive controls.

DECISION MATRIX

DISCUSSION:

Factors	Course of Action	
	1	2
Casualty estimate	+	-
Medical evacuation routes	-	+
Suitable location for medical facilities	0	0
Available EPW facilities	-	+
Suitable CP locations	-	+
Courier and distribution routes	-	+
Effects of attachments and detachments on force cohesion, casualty reporting, and replacement operations	-	+
Residual Risk	+	-

BROAD CATEGORIES

Figure 8. Example decision matrix showing two ways to make comparisons (From U.S. Army, 1997, pp. 5-26)

6. COA Approval

At this step, the commander selects the COA that he believes to "be most advantageous" (U.S. Army, 1997, pp. 5-26). He also refines his intent statement and resource requirements to be in line with the approved COA. A final warning order is issued that reflects the COA selection and new information.

7. Orders Production

The final step in the process results in the production of an OPLAN or OPORD that reflects the COA selected in the previous step. The staff expands the COA into a concept of operations statement that subordinates can use to guide their actions. This statement makes clear "where, when, and how to concentrate combat power to

accomplish the mission in accordance with [the] higher commander's intent" (U.S. Army, 1997, pp. 5-27).

The MDMP is a thorough but typically manpower intensive process. Figure 9 illustrates the various products that are required and produced at each step in the process. Since each step flows naturally into the next, and in turn is dependent on the previous step, time cannot be saved by skipping or eliminating steps in the process. Instead, the process can be shortened by faster creation and evaluation of COAs or reducing the number of COAs created.

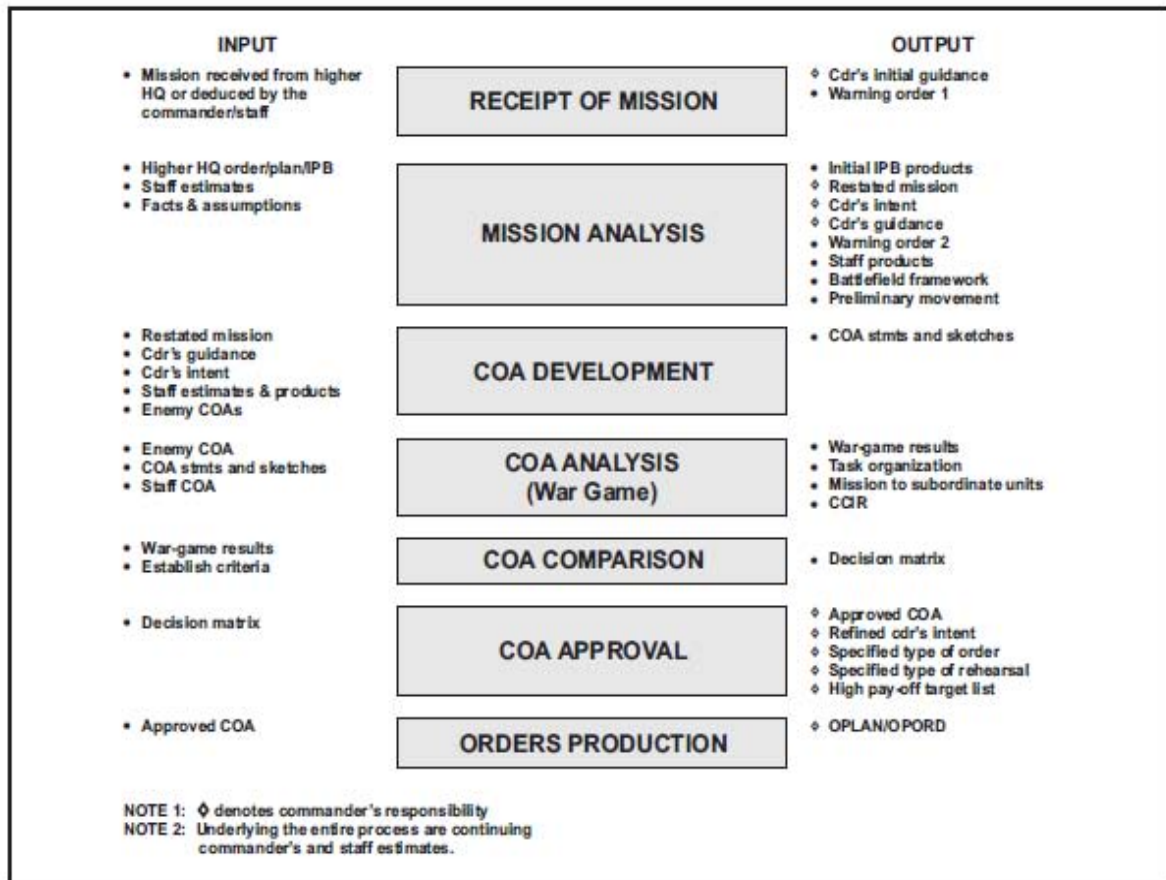


Figure 9. Staff inputs and outputs (From U.S. Army, 1997, p. 4)

C. GENETIC ALGORITHMS

Genetic algorithms were developed in the early 1970s by John Holland in order to mimic the mechanisms that drove natural evolution and to apply them to solving complex problems (Davis, 1991, p. 2). The process of evolution is dependent on chromosomes which act as the "encoding structure of living beings" (Davis, 1991, p. 2). The Handbook of Genetic Algorithms, notes the following features of evolution:

- The process does not operate on living beings but on their chromosomes.
- Natural selection is the determining factor of successful chromosomes. Beings that have more favorable chromosome encodings will reproduce more than those that do not.
- Evolution takes place in reproduction. Recombination and mutation create new chromosomes from two parents in the child.
- Evolution is done without memory. The knowledge of producing successful individuals is contained in the gene pool of a population. (Davis, 1991, pp. 2-3).

These features of evolution highlight some of the requirements and characteristics of genetic algorithms. Genetic algorithms are useful for searching a population of possible solutions for the most appropriate one. Genetic algorithms require that each possible solution in the population be "coded as a finite-length string" (Goldberg, 1989, p. 7). This is usually assumed to be a binary string. Genetic algorithms also require a type of evaluation function to play the role of natural selection. An evaluation, or "fitness," function is the "link between the genetic algorithm and the problem to be solved" (Davis,

1991, p. 4). This function can take any possible solution string and return the value of its fitness. This information is then used by the genetic algorithm to determine reproduction rates. Like evolution, the reproduction process in a genetic algorithm is the main way new encodings are introduced into the population. Parent strings are selected and paired according to their fitness function and mated with each other creating one or more offspring (Haupt, 2004, pp. 41). The offspring represent a recombination of the strings and, as Holland stated in the schema theorem, the patterns (schema) present in the parents. Additionally, some strings may be mutated by having some of their bit values changed. Mutations are important because they "introduce traits not in the original population," (Haupt, 2004, p. 43) and ensure that the algorithm has the ability to generate all the possible encodings of a solution. Finally, the new offspring are introduced into the population and the process repeats itself.

1. Advantages of Genetic Algorithms

Genetic Algorithms are not the only way to solve complex problems, but they do offer several advantages over more traditional methods. For example, consider a function with a solution space shaped like a mountain. The peak of the mountain could be found by having an algorithm choose any random point on it. The algorithm would examine the surrounding points and move to the highest one. This process would repeat until the algorithm returned a location that had no higher surrounding points. This method is called hill climbing (Goldberg, 1989, pp. 3), and

while it is useful for a single mountain, it cannot be used to find the highest point in a solution space that includes multiple peaks. Hill climbing is susceptible to any local maximums in the solution space it is searching. It will find a peak of any mountain in the solution space, but there is no guarantee that the particular peak it returns is the highest one. Figure 10 illustrates the weakness of hill climbing.

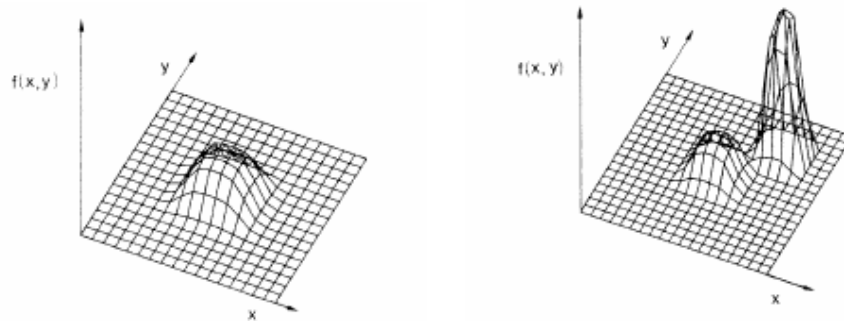


Figure 10. Hill climbing can easily find the peak of the first surface, but it may become caught in the local maximum on the second (From Goldberg, 1989, pp. 3-4)

Unlike hill climbing, a genetic algorithm is able to search solution spaces with multiple local maximums more robustly. In the above mountain range example, a genetic algorithm starts with a population of locations and a fitness function that can return the elevation of any given point. Since points with higher elevations are more likely to be selected and reproduce the average elevation of the population will increase over a number of generations. Additionally, mutations ensure that the entire range of possible locations remains available to the algorithm. The

ability to choose the best point of a population of locations allows the algorithm to move past any local maximum it encounters.

2. The Basic Genetic Algorithm

Given the basic building blocks of a population of encoded solutions, a fitness function, and a reproduction process, a variety of different genetic algorithms can be produced. While the building blocks may change depending on the behavior desired by the developer, each instance of a genetic algorithm follows similar steps.

1. Initialize a population of possible solution strings.
2. Evaluate each string according to the fitness function.
3. Create new strings by mating the current strings. The probability of being selected for mating of any particular string is determined by its fitness score. Strings with higher fitness scores have a higher probability to reproduce.
4. Delete members of the population as needed to maintain desired number of strings.
5. Evaluate the new strings and insert them into the population.
6. If an end-state is reached, return the solution encoded by the string with the highest fitness score; else, repeat process from step 3. (Davis, 1991, p. 5).

These steps represent the most basic implementation of a genetic algorithm. Different techniques for population control, parent selection, and reproduction allow programmers to tailor the performance of the algorithm.

3. Population Controls

The population of a genetic algorithm contains the combined knowledge or explored area of a problem space. There is no set required population size or need for the population to remain constant in the algorithm (Goldberg, 1989, p. 62). However, some instances of the algorithm do mandate maximum, minimum, or even constant levels of population.

Without the removal of some strings, the population size will increase with each generation as new solutions are generated by reproduction. Generational genetic algorithms discard all of the parent strings and form a new population from their children in every generation (Coley, 1999, p. 83). This prevents growth of the population but risks losing many of the best solution strings through failures to reproduce, recombination, and mutation.

To help preserve higher ranking strings through multiple generations, but still maintain a set population size, many genetic algorithms use steady-state reproduction. Steady-state reproduction creates a set number of new strings and removes an equal amount in each generation (Davis, 1991, p. 35). The determination of which strings to remove is usually based on fitness scores. Additionally an algorithm can discard child strings that are duplicates of strings already in the population (Davis,

1991, p. 37). The benefit of removing duplicates is a "more efficient use of our allotted number of chromosomes [strings]" (Davis, 1991, p. 37).

Two other issues that affect the population of a genetic algorithm are elitism and seeding. Elitism entails ensuring that the string with the highest fitness score is carried from one generation to the next (Haupt, 2004, p. 245). This ensures that the most suitable solution discovered so far is not lost due to failure to reproduce or mutation. Seeding is done by inserting "good guesses to the optimum [string] values in the initial population" (Haupt, 2004, p. 249). Additionally, seeding can take place throughout the algorithm by adding good guesses to the population at each new generation.

4. Parent Selection

Parent selection in a genetic algorithm ensures that the strings with the highest fitness scores are given the best chance for reproduction. There are a variety of ways to implement parent selection, with two of the most popular being tournament and roulette wheel selection (Haupt, 2004, p. 41).

a. Tournament Selection

Tournament selection starts by selecting a random sub-set of two or more solution strings from the current population. The string with the highest fitness score in the sub-set is selected for reproduction. This process is repeated until the required number of parent strings have been selected (Haupt, 2004, p. 40).

b. Roulette Wheel Selection

The first step in roulette wheel selection is to calculate the sum of the fitness scores for all of the strings. Next a random number between 0 and the total sum of the fitness scores is generated. Then the fitness scores of the population are added together in sequence and a running total is recorded. The string whose fitness score makes the running total greater than or equal to the random number is selected for reproduction. A new random number is generated to select each subsequent parent. (Davis, 1991, p. 14).

This method of selection gives each string in the population a position on an imaginary roulette wheel proportional to their relative fitness score. The random number represents the ball on the wheel and is more likely to select strings with a greater area or higher fitness score. Table 1 shows a sample population of four strings, their respective fitness scores, and their percentage of the total fitness sum. These percentages are then used to create the roulette wheel shown in Figure 11.

No.	String	Fitness	% of Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100.0

Table 1. A sample population of four strings, their fitness scores, and percentage of the total fitness sum
(From Goldberg, 1989, p. 11)

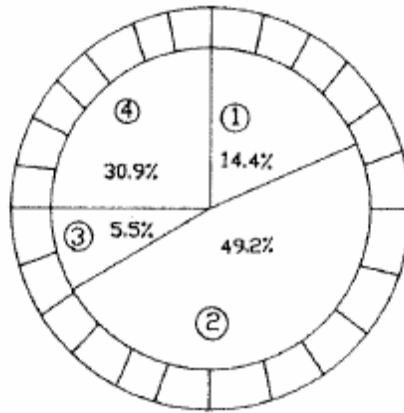


Figure 11. A roulette wheel constructed from the sample population in Table 1 (From Goldberg, 1989, p. 11)

One important consideration for parent selection methods is selective pressure. Selective pressure is the ratio of the probability that the string with the highest fitness score is selected to the probability that the average string is selected (Haupt, 2004, p. 41). If selective pressure is too high, diversity in new generations will suffer. This is caused by the highest ranking string being over-selected for reproduction. In tournament selection, the higher the number of strings in each tournament, the greater the selective pressure. For roulette wheel selection, the presence of super-individuals, strings whose fitness scores are significantly higher than the others, can greatly increase selective pressure (Alba, 2006, p. 1-9).

5. Reproduction and Mutation

Reproduction is the recombination of strings that have been selected to be parents. Most genetic algorithms recombine strings in pairs. The pairing of parent strings can be done randomly, or by ranking potential parents and

having the best available strings mate with each other. Additionally, some genetic algorithms can assign a probability that strings selected for mating with each other will go through the recombination process. The most widespread form of recombination is through crossover.

a. One-point Crossover

The simplest method of recombination, one-point crossover, "occurs when parts of two parent chromosomes are swapped after a random selected point, creating two children" (Davis, 1991, p. 17). An example of one-point crossover can be seen by first creating two parent strings:

Parent A: 0 0 1 1 1 0

Parent B: 1 1 1 0 0 1

A crossover point for the two strings is then selected at random. The characters occurring after the crossover point are swapped between the strings creating two new child strings. If the crossover point for the two strings above came out to be three, then the strings would be divided like this:

Parent A: 0 0 1 | 1 1 0

Parent B: 0 1 1 | 0 1 1

The new child strings would then be:

Child A: 0 0 1 0 1 1

Child B: 0 1 1 1 1 0

b. Schemata

One-point crossover illustrates the importance of schemata in the genetic algorithms. A schema is a "fixed

template describing a subset of strings with similarities at certain defined positions" (Coley, 1999, p. 46). The strings in the above example were constructed from the characters 1 and 0. Schemata require a third character, #, which acts as a placeholder and can equal either 1 or 0.

In the one-point crossover example above, both the parent and child strings contained the schema:

0 # 1 # 1 #

The following schema is only found in Parent B and Child A:

0 # 1 0 1 1

Schemata have two main properties: order and defining length. Order is the number of fixed positions in the schema (Goldberg, 1989, p. 29). In the two schemata above, the first has an order of three and the second has an order of five. Defining length is number of bits between the first and last fixed position in a schema (Goldberg, 1989, p. 29). The first schema has a defining length of four and the second has a length of five.

John Holland's Schema Theorem explains how the manipulation of schemata through reproduction is the critical element of genetic algorithms. Simplified, the theorem states that schemata that are found in strings with above average fitness scores will occur more frequently than those found in below average strings (Davis, 1991, p. 20). Additionally, when using crossover reproduction, the presence of schemata that have a low order, short defining length, and above average fitness scores increases exponentially in each generation (Goldberg, 1989, p. 33).

Holland describes the ability of genetic algorithms to manipulate large numbers of schemata in each generation as intrinsic parallelism (Davis, 1991, p. 20). In the early generations of a genetic algorithm, positive schemata are spread throughout the population, giving many strings parts of a good solution to the problem. Negative schemata will also be spread through the population, meaning strings may also contain parts of a "less optimal" solution. Strings with more positive schemata will be more likely to be selected for reproduction. Their recombination with strings that also have positive schemata, will start to bring the "pieces" of good solutions together. At the same time that positive schemata are being brought together, strings with mostly negative schemata are not as likely to be selected for reproduction. This causes the negative schemata to die out in the population over several generations.

c. Mutation

Mutation is the second way that new strings are added to the population. Unlike recombination, mutation requires only one string instead of a pair. Mutation can be performed on either parent or child strings. Mutations are important in the population because they allow the algorithm to have the possibility of generating schemata that cannot be created through recombination of the current generation. Mutation rates are usually kept low, but can be changed depending on the problem being solved (Coley, 1999, p. 22). High mutation rates can interfere with the algorithm's ability to transfer schemata between generations.

Bit mutation is one example of how a string can be mutated. Each bit in the string has a probability of being replaced based on the mutation rate. A mutation rate of 0.005 means, on average, that five out of every thousand bits will be replaced. Replacement of bits can be done by random selection of a new bit, or by forced swapping of the current bit. Random selection replacement only causes actual mutations in half of the bits picked for mutation. This is due to the possibility that the randomly-selected new bit value will be the same as the old value.

d. Two-point Crossover

A weakness of one-point crossover is that there are some schemata that it cannot combine (Davis, 1991, p. 48). This is usually seen in schemata with fixed positions at the edges of the solution string. An example can be seen in the following two parent strings:

Parent A: 1 1 0 0 0 1 1

Parent B: 0 0 1 1 1 0 0

Parent A contains the schema:

1 1 # # # # 1

One-point crossover is unable to produce this schema in either of the child strings. While it is possible that the schema could appear in the next generation due to mutation, it is not likely unless mutation rates are high.

A technique to overcome this weakness is called two-point crossover. It is similar to one-point crossover, except two random crossover points are generated. The bits between the two crossover points are then swapped between

the two parents. If the crossover points for the two strings above were two and five, then they would be divided like this:

Parent A: 1 1 | 0 0 0 | 1 1

Parent B: 0 0 | 1 1 1 | 0 0

The following child strings would be produced:

Child A: 1 1 1 1 1 1 1

Child B: 0 0 0 0 0 0 0

Note that the schema from above is present in Child A.

6. Convergence

A properly implemented genetic algorithm should see its population become less diverse with each generation as more fit schemata begin to become more prevalent according to the schema theorem. The movement of the population to consisting of essentially similar individuals is called convergence (Davis, 1991, p. 25). Looking at the two solution spaces from the hill climbing example in Figure 10, a genetic algorithm would be expected to produce solution populations that would converge toward the peaks of the mountains over several generations. Convergence is useful because it allows the algorithm to produce higher-quality string populations, and return better solutions.

The less convergence that occurs in a genetic algorithm, the more it resembles a random inspection of all possible solutions. Too much diversity between generations can hinder convergence, and may indicate that there is a problem in the implementation of the genetic algorithm. Generational genetic algorithms and high mutation rates

generally hurt convergence. Crossover methods and population size can also affect convergence (Haupt, 2004, p. 109).

Genetic algorithms that converge too quickly are also not desirable. Convergence after only a few generations may mean that the algorithm is failing to make a reasonable search of all of the possible solution space, and makes the algorithm more susceptible to local peaks or valleys. . Additionally, continuing to produce new generations after convergence is inefficient. Once mutation becomes the main source of new strings in a population, rather than recombination, the algorithm should be stopped (Haupt, 2004, p. 47).

7. Niches

Niches exist as separate subdomains of a particular function (Goldberg, 1989, p. 185). In the example of searching a mountain range for the highest point, each individual mountain could be thought of as a separate niche. Searching for the highest point with a simple genetic algorithm will produce a population of locations centered on the peak of the highest mountain. To produce a population containing representations of the niches around each of the peaks requires modification of the basic genetic algorithm.

One solution, proposed by David Goldberg and Jon Richardson, uses a sharing function to calculate how "similar" an individual string is to other strings. Higher sharing values indicate a string exists in the same general location of several other strings. The string's fitness function is then divided by its sharing value to produce a

degraded fitness score. The degraded fitness score is then used for selecting parents for reproduction in the genetic algorithm (Goldberg, 1989, pp. 191-192). The net effect of the sharing function is that strings that are similar to other strings have their fitness scores lowered, allowing dissimilar strings with relatively high fitness scores a better chance of being selected for reproduction.

Figure 12 shows Goldberg and Richardson's results from a genetic algorithm that does not use sharing, and one that does, when run on a multi-peak function without mutation.

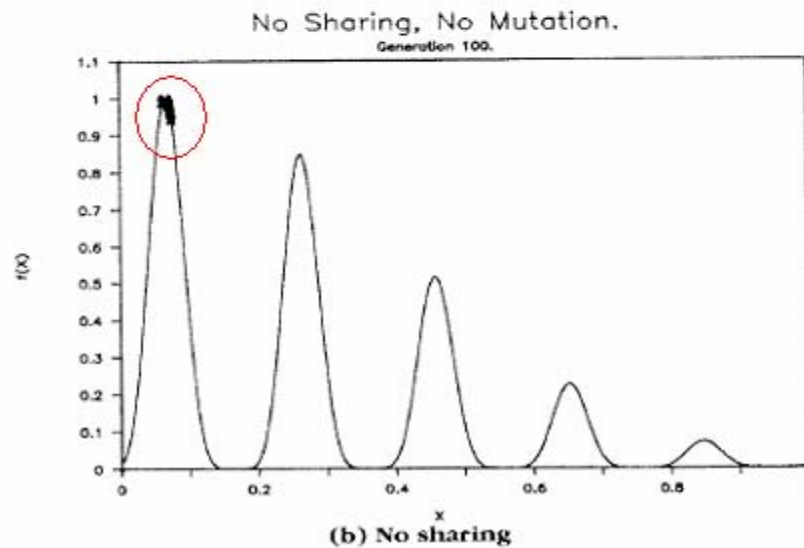
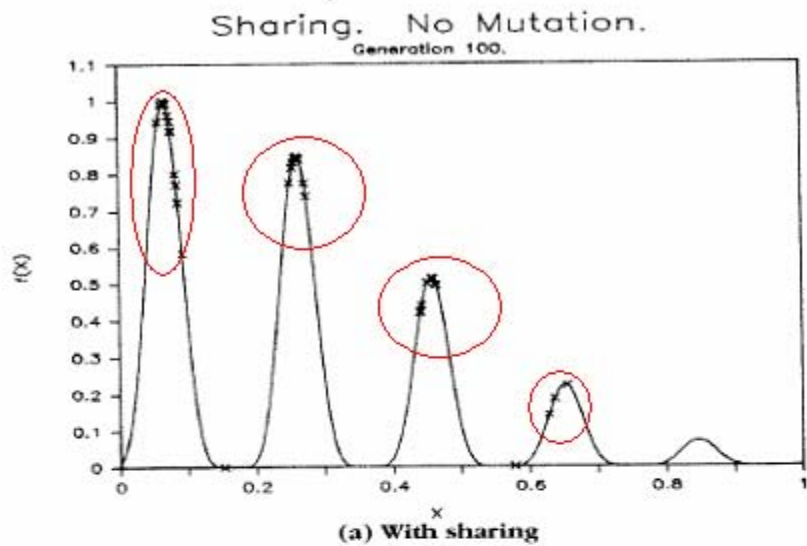


Figure 12. Plots for a value (x) over a multi-peak function, $f(x)$. The potential solution populations at generation 100 for two different genetic algorithms, both searching for the value of x that maximizes $f(x)$ are marked as points on the curve and circled. The first genetic algorithm includes sharing in its fitness function, while the second does not (After Goldberg, 1989, p. 194)

David Goldberg also outlines work by Kenneth De Jong using overlapping populations and crowding factors to implement a way to find niches in a genetic algorithm (1989, pp. 111-116). De Jong uses a variable called a generation gap to differentiate between nonoverlapping and overlapping populations. In nonoverlapping populations, the generation gap equals one, while in overlapping populations it is a value between zero and one. The generation gap represents the fraction of strings that will be selected for breeding in an overlapping population. An equal number of strings are also selected at random to be replaced by the resulting child strings (Goldberg, 1989, pp. 111).

De Jong's crowding model uses an overlapping population and a new parameter called the crowding factor. In the crowding model, each newly-created child string replaces an existing string in the population. To select the string that will be replaced by a new child string, a sub-set of strings in the current population is created. The crowding factor equals the number of strings picked at random in the sub-set. The string in the sub-set that is most like the child string, based on a bit-by-bit similarity count, is replaced. (Goldberg, 1989, p. 116)

Since new strings are more likely to replace strings that are similar, strings with unique characteristics (but with possibly lower fitness scores) have a greater chance to be present in future generations. This is similar to removing strings that are duplicates from the population.

D. SUMMARY

This chapter described the steps in the MDMP and the process of creating and evaluating COAs. This chapter also showed how genetic algorithms have the capability to solve complex problems, provided solutions can be encoded in some type of string. The following chapter will show how the BBE is able to encode COAs and apply a genetic algorithm to rapidly create and wargame thousands of computer-generated COAs.

THIS PAGE INTENTIONALLY LEFT BLANK

III. THE BTRA-BC BATTLE ENGINE

A. INTRODUCTION

The BTRA-BC Battle Engine (BBE) uses products created in the mission analysis step of the MDMP to allow planners to create COAs that can be evaluated and evolved in a genetic algorithm. This chapter describes some of the basic inputs, features, and functions of the BBE. In addition, the user controls of the genetic algorithm are defined.

Much of the information in this chapter comes from the "Cognitive Amplification for Contextual Game-Theoretic Analysis of Military Courses of Action," which is an invention disclosure for the BBE written by Jerry Schlabach and Eric Nielsen.

B. MISSION ANALYSIS AND INITIAL INPUTS

Mission, Enemy, Terrain, Troops and Time (METT-T) are some of the main considerations in the MDMP. Data on all of these topics are gathered and studied during the IPB process in mission analysis. METT-T data are some of the initial inputs to the BBE. The BBE Mission Building menu, seen in Figure 13, is where this information is entered.

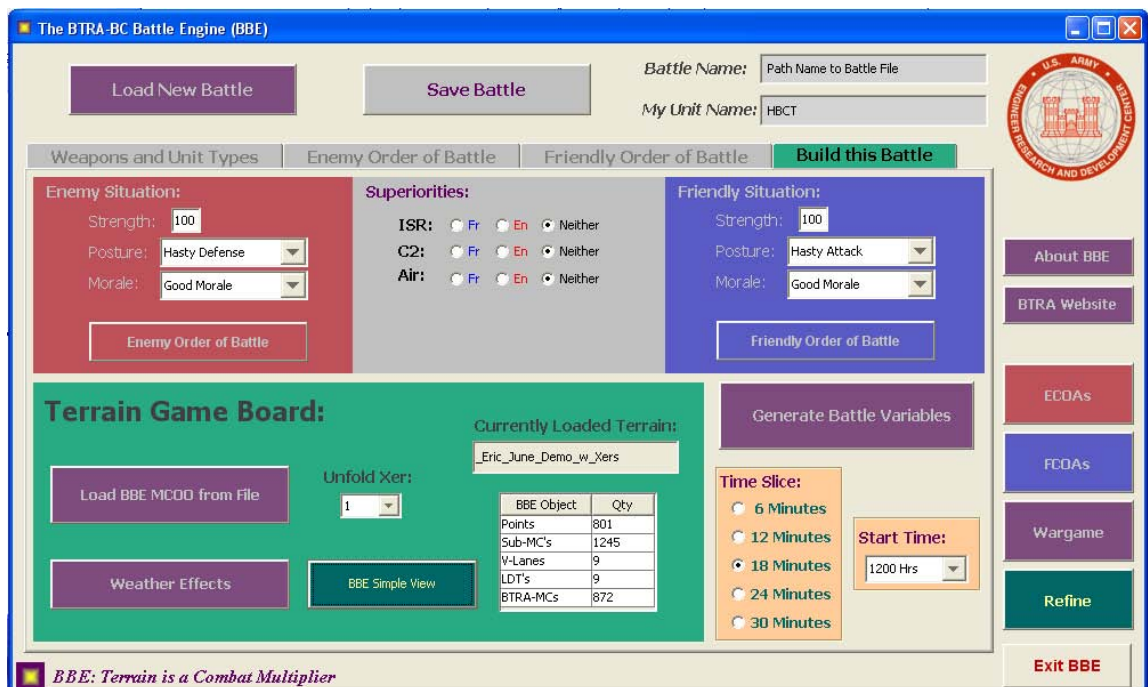


Figure 13. The BBE Mission Building menu

The Mission Building menu acts as the main hub for all of the BBE functions and is the initial screen presented to the user. This menu allows the user to load and save battle scenarios. The buttons on the left side lead to ECOA and FCOA construction menus and the Wargame menu. Tabs in the middle section allow the user to enter intelligence developed during the IPB. Additionally, the user can enter the mission start and time slice to be used for simulation.

1. Terrain

The MCOO, traditionally produced during the IPB, gives mission planners an abstracted idea of the terrain in the operational area. It outlines avenues of approach (AA) to be used by attacking forces and the lines of defensible terrain to be used the defending forces. In a yellow

sticky drill, the MCOO would be the map that the units are moved across. In the BBE, the MCOO is loaded at the bottom of the Mission building menu as seen in Figure 14.

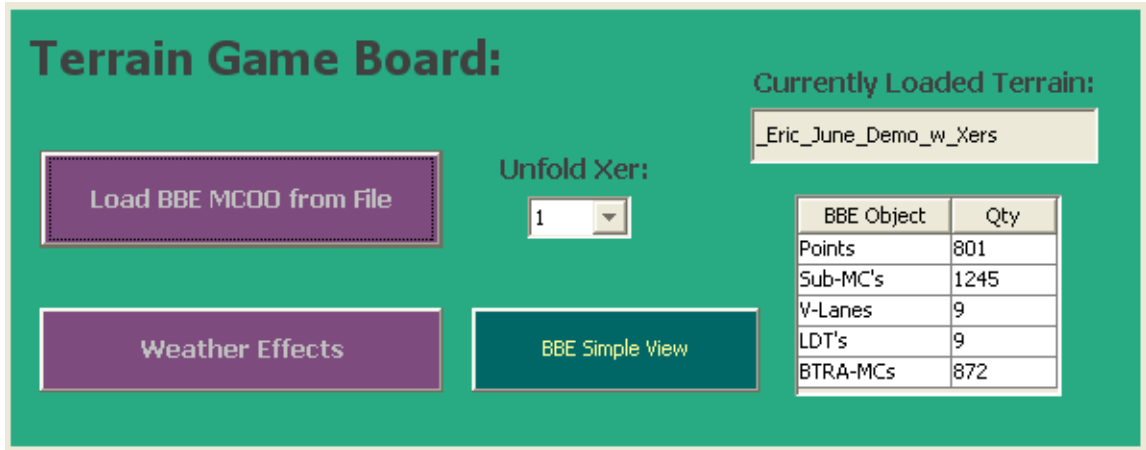


Figure 14. The MCOO loader in the Mission Building menu

The BBE uses a separate program to create MCOOs for battle scenarios. The MCOO-Maker uses an abstraction of the operational terrain called a "Braswell Index," to construct mission game boards. The Braswell Index creates a network of mobility corridors around obstacles. The index includes characteristics that outline combat multipliers for attacking and defending forces in each mobility corridor (Schlabach, 2009, pp. 8-9). An example of a Braswell Index can be seen in Figure 15.

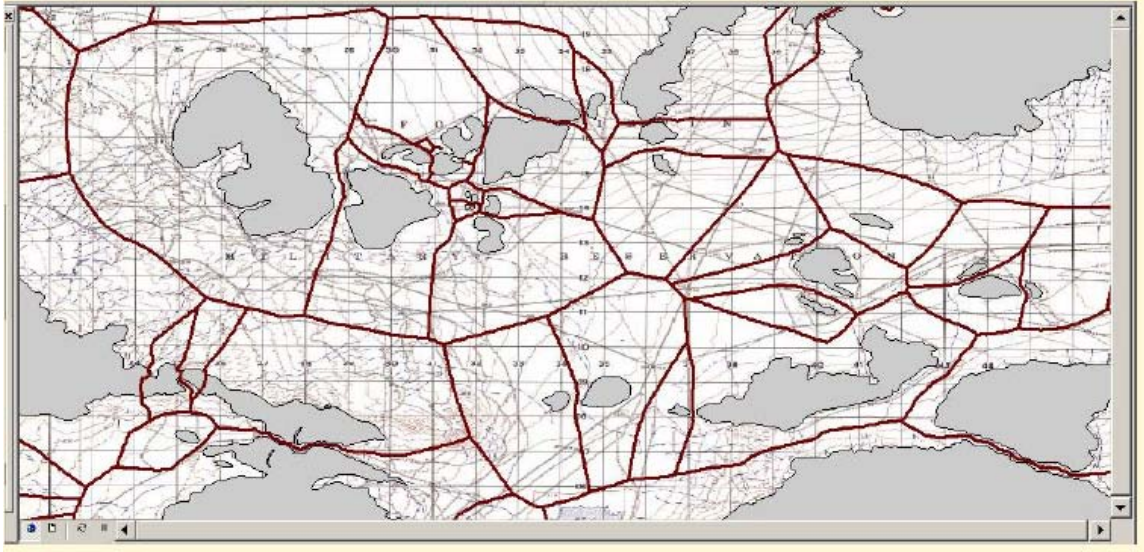


Figure 15. An example of a Braswell Index. The lines
 "bisect the Mobility Corridors between the obstacles"
 (From Schlabach, 2008, p. 7)

The game boards created by the MCOO-Maker contain enough terrain data to enable environmental combat multipliers, but can still be loaded into the computer's basic memory. This allows for faster battle simulations than if the game board was loaded from the computer's hard drive (Schlabach, 2009, p. 9).

The BBE game boards also contain information that guides the movement of offensive units and the placement of defensive units. For the attacking units, the game board has a set of Virtual (V) Lanes that are analogous to AAs on a typical MCOO. V-Lanes are a series of routes across the game board extend from a unit's start point to its objective. An example of a set of V-Lanes can be seen in Figure 16. V-Lanes are logically parallel to each other and act as the guide for offensive movement in the BBE simulations. Additionally, V-Lanes contain information

that indicates the type of attacking formations they can support (Schlabach, 2009, p. 10).

For defensive units the game board identifies lines of defensible terrain (LDT). LDTs are made from "neighboring mobility corridors upon which a coherent defense can be based" (Schlabach, 2009, p. 10). LDTs usually intersect V-Lanes and act as placement points for defending units. An example of an LDT can be seen in Figure 17.

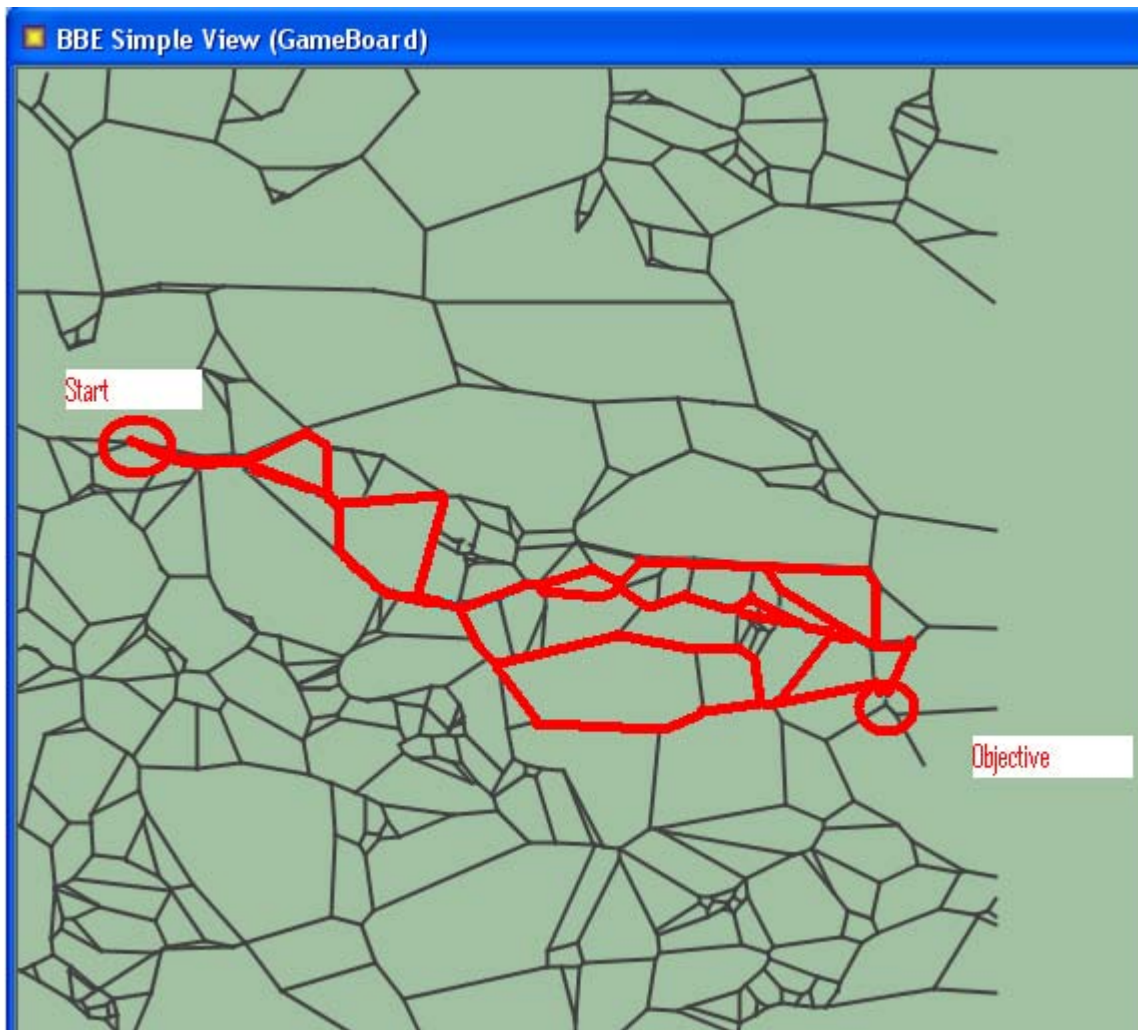


Figure 16. BBE game board with several V-Lanes highlighted. The mission start point and objective have also been annotated.

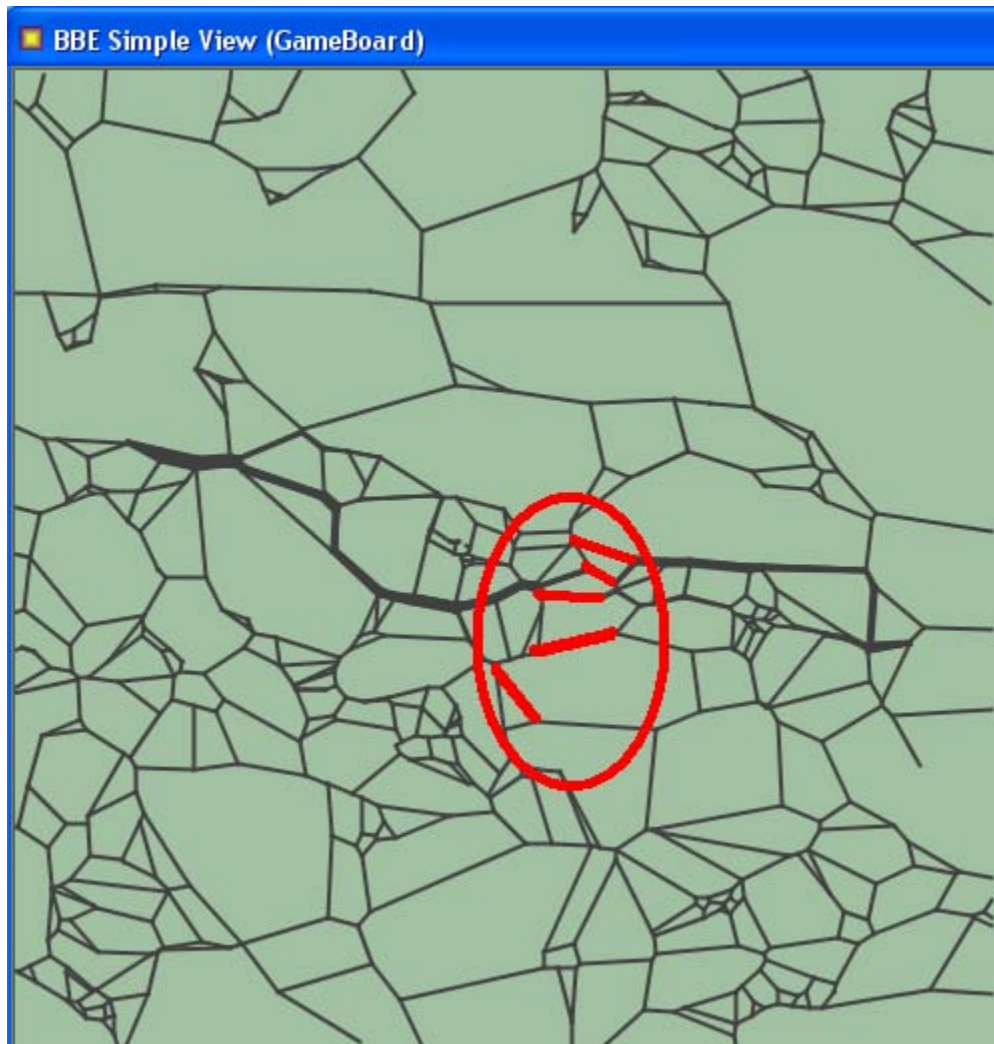


Figure 17. BBE game board with a LDT circled. Note how it intersects the V-Lanes seen in Figure 16.

2. Order of Battle

The next elements entered into the BBE are the friendly and enemy orders of battle. In the yellow sticky drill, these would be the post-it notes that are moved across the MCOO. Data to build these units would be collected during the IPB process in mission analysis. Like the Terrain loader, the BBE relies on an external program, the BBE Weapons Assessment and Calculation Tool (B-WACT), to provide weapon data. Seen in Figure 18, the B-WACT

allows mission planners to “develop a basic relative combat power for individual weapons and weapon systems that aggregate weapons” (Schlabach, 2009, p. 11). The combat power ratings are based on the “Quantitative Judgement Method of Analysis” (QJMA) developed by Colonel Trevor N. Dupuy (Schlabach, 2009, p. 11). A list of these weapons can be loaded in the Weapons and Unit Types tab in the Mission Building menu seen in Figure 19. Mission planners can also group weapon systems together to form both friendly and enemy units in this tab.

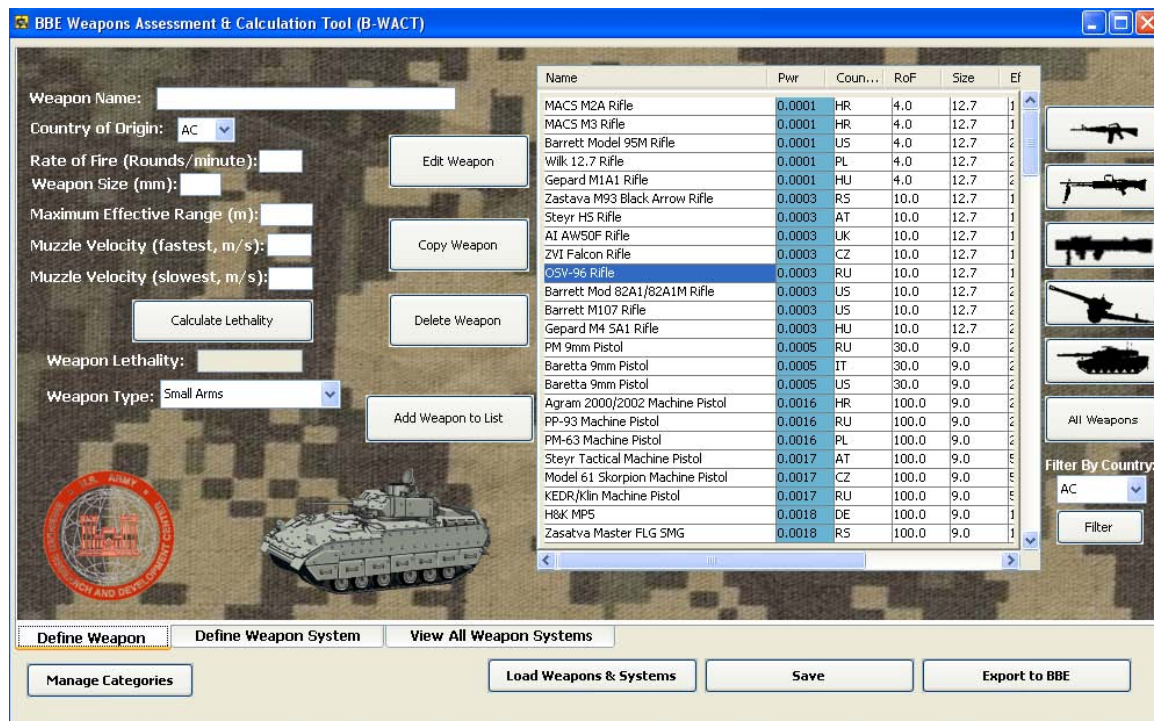


Figure 18. The B-WACT is used to build units for use in the BBE

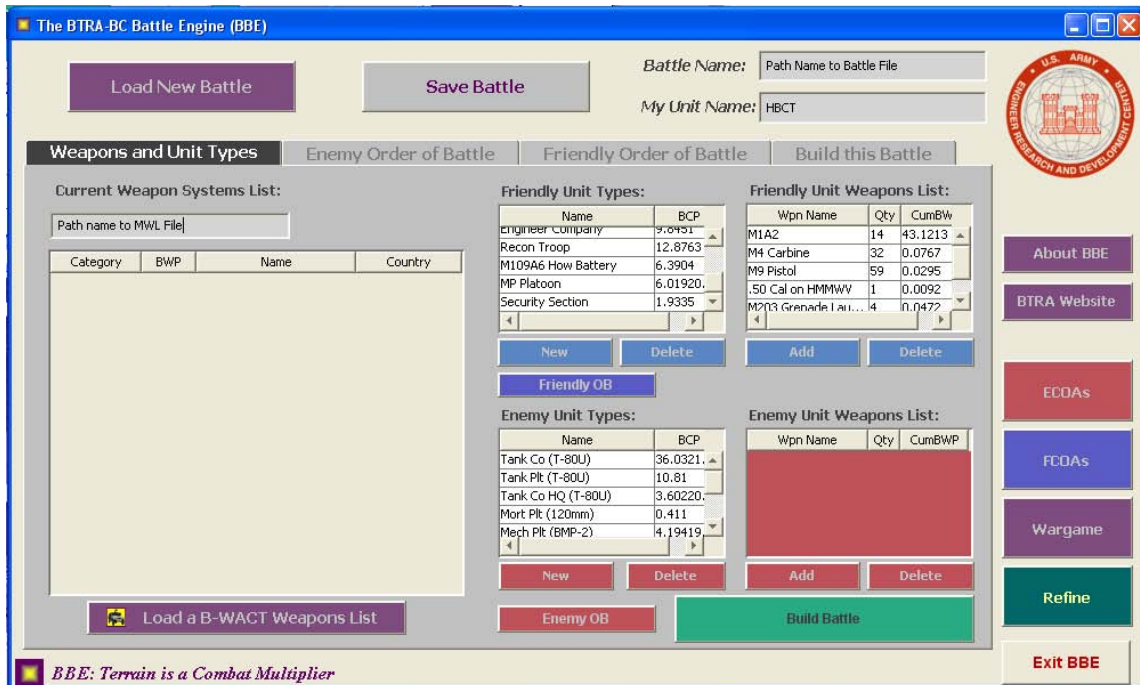


Figure 19. The Weapons and Units Types tab. In this menu, mission planners can load B-WACT weapons lists and build units

Once unit types have been built, the mission planner can begin to construct orders of battle (OB) for both the friendly and enemy forces. Both sets of forces have their own respective OB building tabs in the Mission Building menu shown in Figure 20 and 21. In these tabs, units are grouped together to form subordinate commands that are then used in COA construction. Previous OBs can also be loaded, or current OBs saved, for later use. Each subordinate is given a Base Power score, which is determined by the basic combat powers of each of its units.

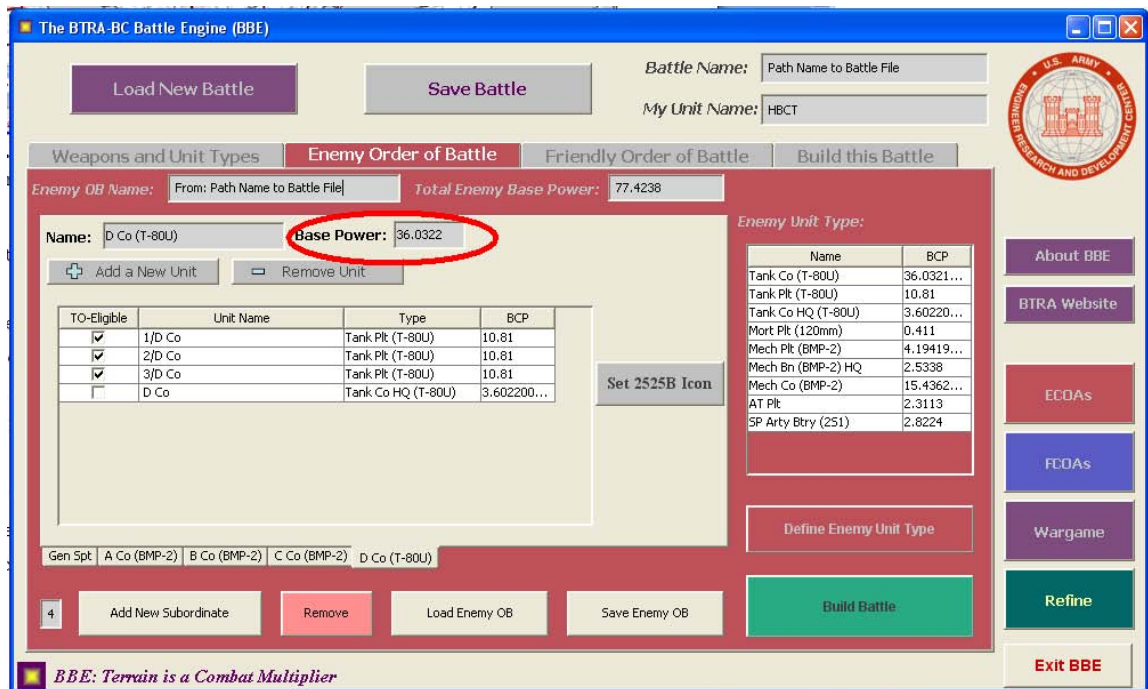


Figure 20. The Enemy OB tab. The base power of the subordinate command is circled in red

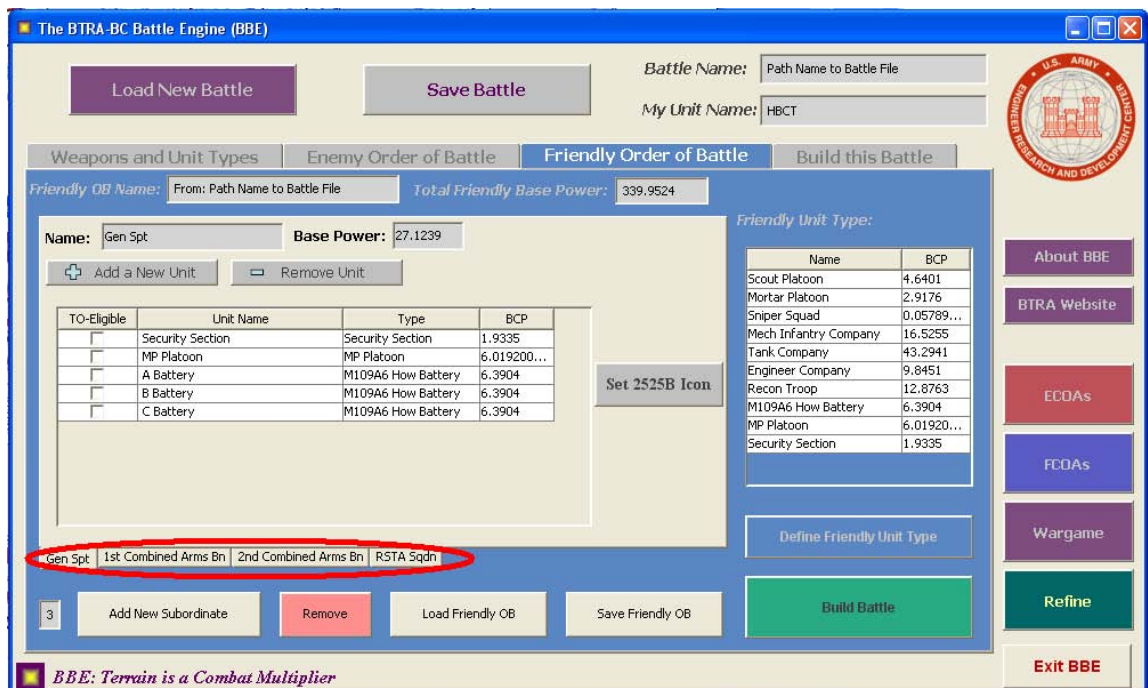


Figure 21. The Friendly OB tab. The subordinate command tabs are circled in red

3. Postures and Morale

The final inputs on the Mission Building menu deal with the postures and morale of the respective forces and act as combat multipliers in the battle simulation. Mission developers can adjust the strength and overall morale of forces to reflect conditions as needed. Morale can be set to five different levels ranging from "Excellent" to "Panic." Force postures are set based on a list that includes hasty attack, prepared attack, fortified defense, hasty defense, prepared defense, delay/withdraw, and movement to contact. Each of these settings "influence[s] the attrition calculations [in the simulation]" (Schlabach, 2009, p. 12). Additionally, the mission planner can set superiorities for each side in Intelligence, Surveillance, and Reconnaissance (ISR), Command and Control (C2), and air power. These superiorities represent advantages that the respective forces may have in communication, intelligence, and control of the operational air space. The superiorities settings act as multipliers that affect unit attrition calculations in the simulations combat model. Figure 22 shows the posture, morale, and superiorities controls.

The image shows a screenshot of the Mission Building menu, divided into three main sections: Enemy Situation (red background), Superiorities (grey background), and Friendly Situation (blue background).

Enemy Situation:

- Strength: 100
- Posture: Hasty Defense (dropdown menu)
- Morale: Good Morale (dropdown menu)
- Button: Enemy Order of Battle

Superiorities:

- ISR: ☐ Fr ☐ En ☒ Neither
- C2: ☐ Fr ☐ En ☒ Neither
- Air: ☐ Fr ☐ En ☒ Neither

Friendly Situation:

- Strength: 100
- Posture: Hasty Attack (dropdown menu)
- Morale: Good Morale (dropdown menu)
- Button: Friendly Order of Battle

Figure 22. Controls for force postures, morale and superiorities

C. COA CREATION

After the terrain, orders of battle, and postures have been set, a mission planner can begin creating COAs. Both ECOAs and FCOAs have their own respective creation interfaces, shown in Figures 23 and 25. In these examples the FCOA creation menu contains offensive COA variables, while the ECOA has defensive variables (the attacker-defender roles can be reversed if needed). The top part of both menus contains a list of prospective COAs that have already been created, along with a short description of the COA. Each ECOA created also has a user-provided value for the relative probability that the enemy will use that particular COA. The BBE also provides a visualization of the COA setting, as seen in Figures 24 and 26.

FCOA Development

FCOA Nominees:

Visualize +

Name	Description
Initial Attack	1 Abreast Formation: !!0, !0F, 0F; No Internal Boundaries.
3 wide attack	3 Abreast Formation: !!0, !1, 2; Boundaries = 2, 5.

Offensive COA Variables:

Num Abreast: 3 Unit Formation: 0, 1, 2 Unit Boundaries: 2, 5 Unit Assignments: 1, 2, 3 Attack On Line: Yes Priority of GS, by formation slot: 1, 2, 3 Severity of GS, by formation slot: 34, 33, 33

Subordinate:

#	Name
1	1st Combined Ar...
2	2nd Combined Ar...
3	RSTA Sqdn

Bypass Criteria:

Bn: Withdrawal Criteria: 95.0 F&S or Reserve: Follow & Spt Reserve Lane: 0 Reserve Threshold: 95.0 Reserve Guidance: Stay in Lane Reserve Lag Distance: 0.5

T.O.-able Units:

#	Name
1	A Troop
2	B Troop

Subordinate:

#	Name
1	1st Combined Ar...
2	2nd Combined Ar...
3	RSTA Sqdn

Control Measure: L. Bndry: 3 R. Bndry: 5 Internal Attack Online: Yes Upon Penetration: Stay At OBJ: Stay Stutter Start: 0.0 Hrs

Figure 23. The FCOA creation menu with offensive COA variables

COA variables are used to construct the bit strings that drive the simulation in wargaming, and can be manipulated in the genetic algorithm. There are three types of COA variables in both offensive and defensive COAs. The first type is total unit variables. These variables are found on the left side of the COA creation menu and affect all units. Total unit variables determine lanes of movement or responsibility, as well as formations and ground support priority. The next type is subordinate unit variables. These variables are found in the middle of the menu and are set for each individual unit. The final type of COA variable deals with Task Organizable (TO) units. TO units can be assigned to any of the subordinate commands in the order of battle. The assignment of TO units is done on the right side of the COA creation menu. While offensive and defensive COAs share many of the same traits, there are some significant differences.

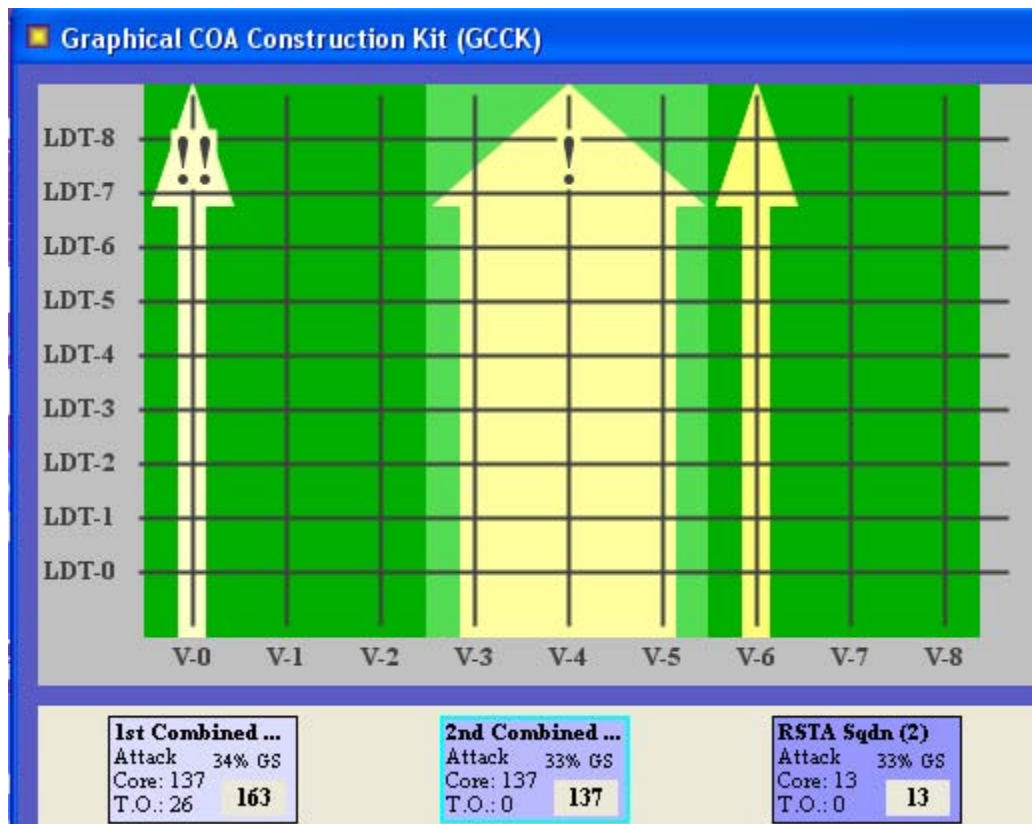


Figure 24. Graphical visualization of the FCOA displayed in Figure 23. Since this is an offensive COA, units are assigned V-Lanes to define their movement. The exclamation points indicate priority for GS. The values underneath the unit names display the units' percentage of general support, the units' core combat power and the combat power of any subordinate forces Task Organized (TO) to the unit. Each unit's total combat power is displayed in bold

1. Offensive COA Variables

The Offensive COA variable set defines the movement of the attacking force across the game board. The Invention Disclosure document by Schlabach and Nielsen defines the following variables.

- **Number Abreast (Num abreast)**—The number of columns used by an attacking force. This value is dependent on available V-Lanes and subordinate units.
- **Unit Formation**—Outlines the position of each unit based on the Number Abreast and number of subordinate units. For example, a value of (0, 1, 1) indicates that there are three subordinate units in two columns. The first unit is in column "0," while the second and third units are in column "1."
- **Unit Boundaries**—Defines the V-Lane boundaries of the unit columns. In Figure 23, a three abreast formation is given the boundaries (2, 5). This means that the first column will advance along V-Lanes 0-2, the second column will use V-Lanes 3-5, and the third column will use the remaining V-Lanes. The illustration of this is seen in Figure 24.
- **Unit Assignments**—Assigns Subordinate units to particular formation slots. The numbers correspond to the Subordinate selection panel in the middle of the FCOA creation menu. A value of (2, 3, 1) would assign the second subordinate to the first formation slot, the third subordinate to the second slot, and the first subordinate to the third slot.

- **Priority of General Support (GS)**—Determines the priority of GS by formation slot. In the visualization of the COA, higher priority is indicated by increased exclamation points as seen in Figure 24.
- **Severity of GS by Formation Slot**—Shows the percentage of GS allocated to each formation slot.
- **Left and Right Boundaries (L. Bndry, R.Bndry)**—These controls restrict subordinate units' movements to particular V-Lanes. In Figure 25, the unit in the second column is given wider left and right boundaries than the other units.
- **Upon Penetration**—Describes unit actions after it penetrates an enemy defensive position. This variable is not completely implemented in the battle simulations but is expected in subsequent versions of the BBE.
- **At Objective (OBJ)**—Describes unit actions once it reaches its objective. This variable is not completely implemented in the battle simulations but is expected in subsequent versions of the BBE.
- **Stutter Start**—Sets a delay for initial unit movement.

- **Bypass Criteria**—Determine the level of defensive force that will be bypassed once the LDT has been penetrated. Ranges go from squad (SQD) to battalion (BN).
- **Withdrawal Criteria**—Indicates the unit strength level at which the unit will withdraw from combat. Expressed as a percentage of overall strength.
- **Follow on Support (F&S) or Reserve**—Used only if the subordinate unit is in a reserve slot in the formation.
- **Reserve Lane**—Determines the V-Lane for a reserve unit.
- **Reserve Threshold**—Determines the subordinate strength level at which it will employ its reserve unit.
- **Reserve Guidance**—Directs the action of the reserve unit. Reserve units can stay in assigned V-Lanes or be set to attack either the "best-dent," "best-hole" (penetration of defense), or "first-hole." This variable is not completely implemented in the battle simulations.
- **Reserve Lag Distance**—This variable controls the distance that a subordinate unit assigned a reserve task will stay behind the unit it is supporting. (Schlabach, 2009, p 18-20)

2. Defensive COA Variables

The defensive COA menu has several variables that are the same as in the offensive COA menu. Number abreast, formation assignments, boundaries, GS priority and severity, and withdrawal criteria are also in the defensive COA variable set. Unlike offensive COA creation, the defensive COA set focuses on unit placement, not unit movement. The following variables are exclusive to the defensive COA menu.

- **Anchor LDT**—Defines the LDT that will be defended. Defensive units will be placed on the mobility corridors that make up the LDT. In Figure 26, the anchor LDT is LDT-5. Units are responsible for defending sections of V-Lanes as they intersect LDT-5.
- **Anchor Line Setback**—Sets the distance a unit will move back from the anchor LDT.
- **Reinforce Policy**—Determines which neighboring units a non-attacked subordinate will reinforce.
- **Delay or Reserve**—Determines if a non-main line defense unit is set in a Delay or Reserve mission. In the example COA in Figure 25 and Figure 26, D Company is set as a reserve unit.
- **Delay Depth**—Distance behind the anchor LDT for a unit ordered to delay.
- **Reserve Lag Distance**—Distance behind the anchor LDT for a unit set to reserve. (Schlabach, 2009, pp. 15-17)

ECO Development

ECO Nominees (IPB):

Visualize + Suggest Coarse ECOA's

Name	Rel.Prob	Description
Balanced Defense - Tk Co Reserve	1.4	3 Abreast Formation: !0, !!1, 2, 1R; Anchor
Balanced Defense - CA Companies	1.5	3 Abreast Formation: !0, !!1, 2, 1R; Anchor
Forward Defense - Tk Co Reserve	1.1	3 Abreast Formation: !0, !!1, 2, 1R; Anchor
Strong Right	1.2	3 Abreast Formation: !!0, !1, 2, 1R; Anchor
Strong Left	1.2	3 Abreast Formation: 0, !!1, !2, 2R; Anchor

Defensive COA Variables:

Num Abreast: 3 Unit Formation: 0, 1, 2, 1

Unit Boundaries: 2, 5

Unit Assignments: 1, 2, 3, 4

Anchor LDT: 5

Priority of GS, by formation slot: 2, 1, 3, 4

Severity of GS, by formation slot: 25, 25, 25, 25

Subordinate:

#	Name
1	A Co (BMP-2)
2	B Co (BMP-2)
3	C Co (BMP-2)
4	D Co (T-80U)

Control Measures:

L. Bndry: 0 R. Bndry: 0

Anchor Line Seback: 0

Reinforce Policy: Neither

Withdrawal Criteria: 75.0

Delay or Reserve: Reserve

Delay Depth: 0.5

Reserve Lag Distance: 2.0

Reserve Threshold: 95.0

Reserve Guidance: Option-A

Reserve Lane: 0

Upon Penetration: Stay

T.O.-able Units:

#	Name
1	1/D Co
2	2/D Co
3	3/D Co

Subordinate:

#	Name
1	A Co (BMP-2)
2	B Co (BMP-2)
3	C Co (BMP-2)
4	D Co (T-80U)




Figure 25. The ECOA creation menu with defensive COA variables

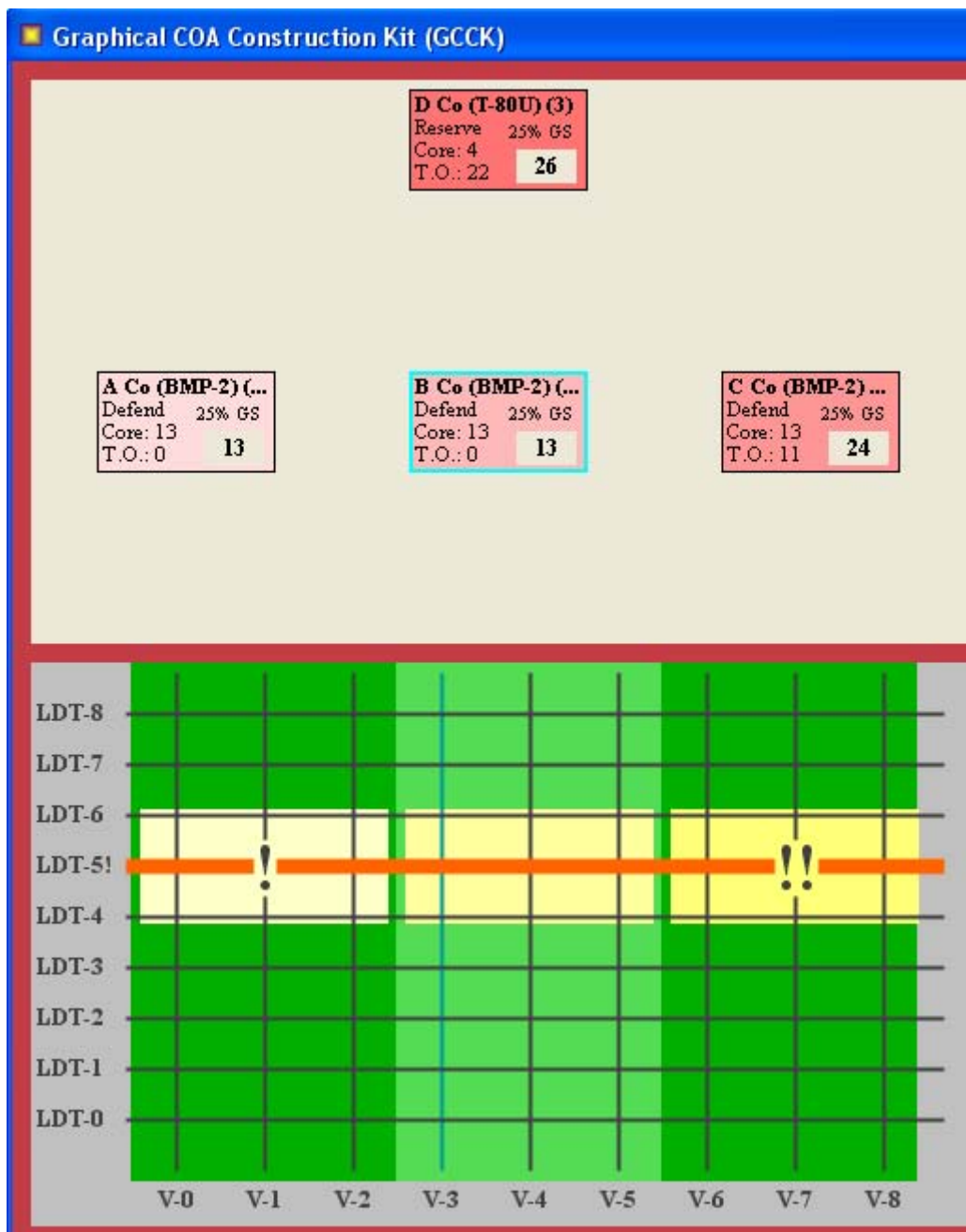


Figure 26. Graphical visualization of the ECOA displayed in Figure 25. In this defensive COA, units are positioned at LDT-5. Each unit is given a range of V-Lanes that they are responsible for blocking. D Company is set to act as reserves for B Company

D. WARGAMING

Once a suitable set of ECOAs and FCOAs has been constructed, the mission planner can move on to the Wargame menu seen in Figure 27. From this menu, the mission planner can set the desired end state, visualize a sample battle between a FCOA and an ECOA, and obtain evaluation scores for the FCOA set.

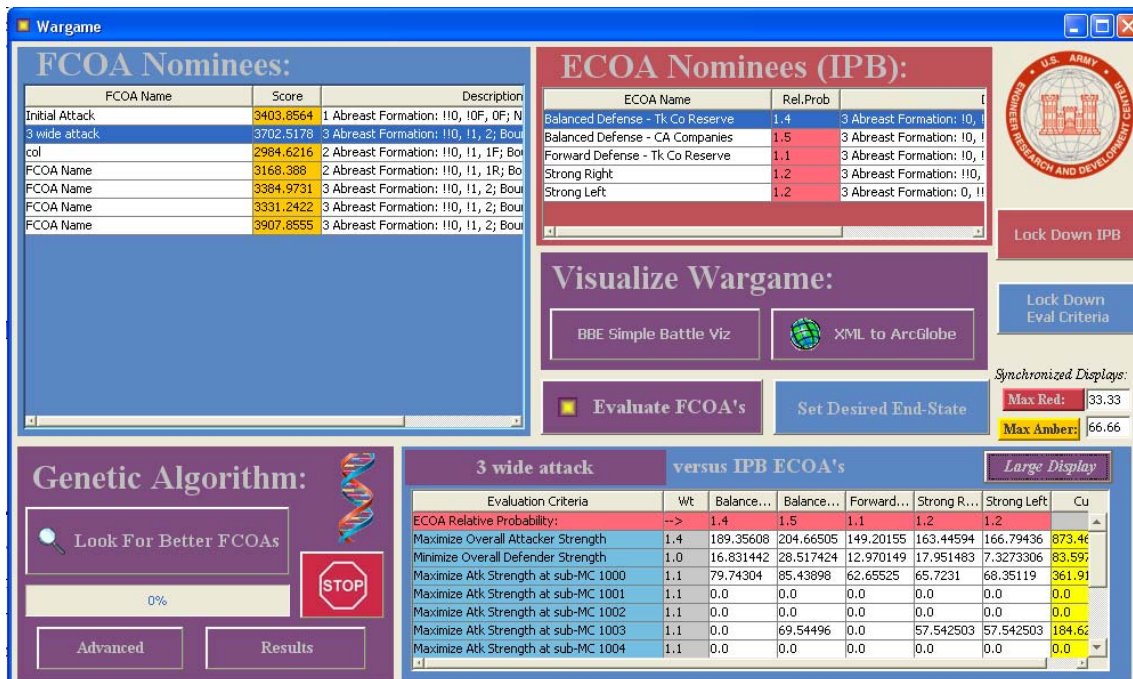


Figure 27. The Wargame menu

1. Desired End State

The desired end state (DES) in the BBE reflects the commander's intent that is created during the MDMF. Commanders can use the DES menu, seen in Figure 28, to "select criteria that reflect how he [or she] would like the battlefield to 'look' at the end of a successful mission" (Schlabach, 2009, p. 28).

Set Desired Battle End State; BTRA-BC Battle Engine (BBE)

Total Evaluation Criteria:

(This is the Working set, not the active Wargame set)

Weight	Name
1.0	Maximize Overall Attacker Strength
1.0	Minimize Overall Defender Strength
1.0	Maximize Atk Strength at sub-MC 1000
1.0	Maximize Atk Strength at sub-MC 1001
1.0	Maximize Atk Strength at sub-MC 1002
1.0	Maximize Atk Strength at sub-MC 1003
1.0	Maximize Atk Strength at sub-MC 1004
1.0	Maximize Atk Strength at sub-MC 1005
1.0	Maximize Atk Strength at sub-MC 1006
1.0	Maximize Atk Strength at sub-MC 1007
1.0	Maximize Atk Strength at sub-MC 1012

Copy from Wargaming Submit to Wargaming

Overall Unit Criteria Candidates:

☒ Maximize Attacker End % Strength
☐ Minimize Attacker End % Strength
☐ Maximize Defender End % Strength
☒ Minimize Defender End % Strength

Time Criteria Candidates:

☐ Maximize Battle Time
☐ Minimize Battle Time

Specific Unit Criteria Candidates:

Attacker: ☒ Max ☐ Min Defender: ☐ Max ☒ Min

Name	Name
<input type="checkbox"/> 1st Combined Arms Bn	<input type="checkbox"/> A Co (BMP-2)
<input type="checkbox"/> 2nd Combined Arms Bn	<input type="checkbox"/> B Co (BMP-2)
<input checked="" type="checkbox"/> RSTA Sqdn	<input type="checkbox"/> C Co (BMP-2)
	<input type="checkbox"/> D Co (T-80U)

☐ Maximize Attacker's Uncommitted Reserve End Strength (%)
☐ Minimize Attacker's Uncommitted Reserve End Strength (%)
☐ Maximize Defender's Uncommitted Reserve End Strength (%)
☐ Minimize Defender's Uncommitted Reserve End Strength (%)

TBD Criteria Candidates:

Mobility Corridor Criteria Candidates:

Max Atk	Min Atk	Max Def	Min Def	Sub-MC Key	V-Lane Index	V-Lane Key	MC Index in V-Lane
false	false	false	false	163	0	1	0
false	false	false	false	210	0	1	1
false	false	false	false	245	0	1	2
false	false	false	false	279	0	1	3
false	false	false	false	291	0	1	4
false	false	false	false	346	0	1	5
false	false	false	false	365	0	1	6
false	false	false	false	401	0	1	7

BBE Simple View (MC Selection for Desired End State)

Figure 28. The Desired End State menu

End state criteria can include total mission time, overall force levels, unit force levels, and unit strength on particular mobility corridors. The current list of selected criteria is displayed in the top left of the DES menu. Each individual criterion is given a weight factor that reflects its relative importance to the commander.

2. Battle Visualization

The BBE also allows planners to visualize sample COA engagements as seen in Figure 29. Battle visualization helps the mission planner see how units are moving across the V-Lanes and confirms desired placement of defensive forces. Planners can also track a unit's status during a particular COA.

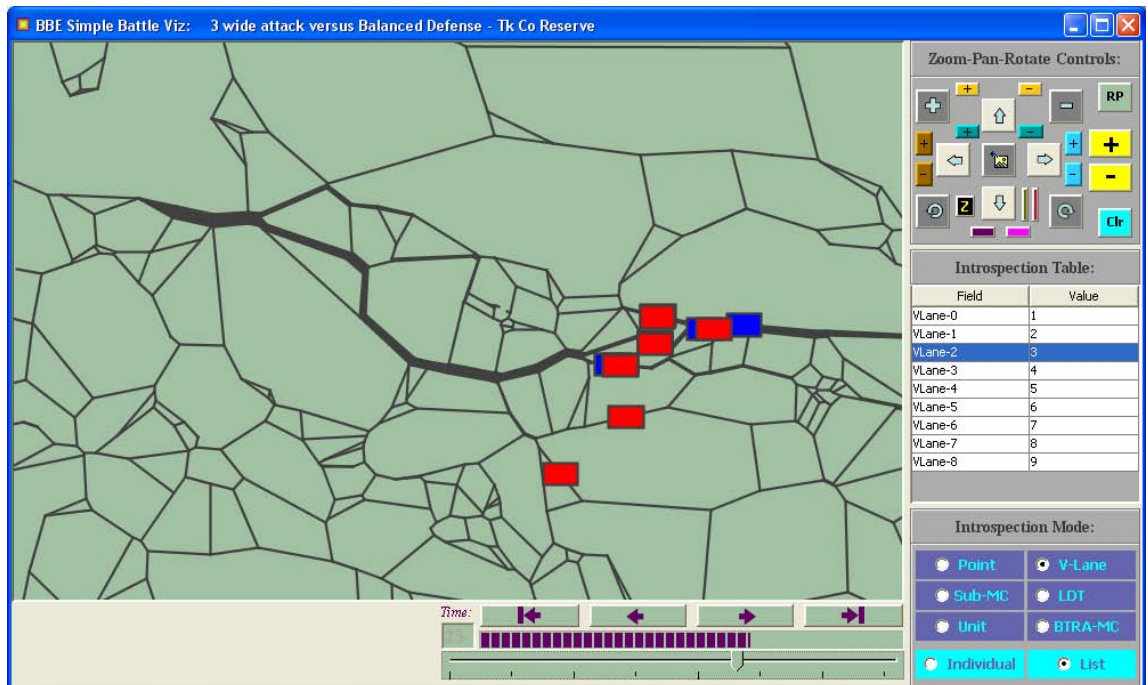


Figure 29. The Battle visualization screen. Controls at the bottom of the screen allow the user to advance the simulation. Unit information and MC data are displayed to the right of the MCOO

3. COA Evaluations

The BBE evaluates battles using a modified Lanchester equation with modified Dupuy QJMA coefficients (Schlabach, 2007, p. 3). Combat occurs anytime a defensive unit and an offensive unit enter the same mobility corridor. Attrition rates are determined by the combat strength of each unit, terrain multipliers in the mobility corridor, and the mission postures and morale set earlier. Battle outcomes are deterministic, so the same FCOA and ECOA battles will produce the same result in each battle.

Each FCOA is evaluated against the entire set of ECOAs using the desired end-state variables, producing an evaluation matrix seen in Figure 30. The individual scores of the FCOA against each ECOA are weighted according to the

user-assigned probability that the enemy will use the ECOA. The scores are then summed to produce the FCOA's total score.

Evaluation Criteria	Wt	Balanced Defense - ...	Balanced Def...	Forward Defen...	Strong R...	Strong Left	Cum
ECOA Relative Probability:	-->	1.4	1.5	1.1	1.2	1.2	
Maximize Overall Attacker Strength	1.4	189.35608	204.66505	149.20155	163.44594	166.79436	873.46295
Minimize Overall Defender Strength	1.0	16.831442	28.517424	12.970149	17.951483	7.3273306	83.59783
Maximize Atk Strength at sub-MC 1000	1.1	79.74304	85.43898	62.65525	65.7231	68.35119	361.91156
Maximize Atk Strength at sub-MC 1001	1.1	0.0	0.0	0.0	0.0	0.0	0.0
Maximize Atk Strength at sub-MC 1002	1.1	0.0	0.0	0.0	0.0	0.0	0.0
Maximize Atk Strength at sub-MC 1003	1.1	0.0	69.54496	0.0	57.542503	57.542503	184.62997
Maximize Atk Strength at sub-MC 1004	1.1	0.0	0.0	0.0	0.0	0.0	0.0
Maximize Atk Strength at sub-MC 1005	1.1	0.0	0.0	0.0	0.0	0.0	0.0
Maximize Atk Strength at sub-MC 1006	1.1	0.0	0.0	4.759064	0.0	0.0	4.759064
Maximize Atk Strength at sub-MC 1007	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Maximize Atk RSTA Sqdn % of TO'ed Start Strength	1.5	195.21951	191.31525	157.71555	169.51662	169.58273	883.3496
Minimize Def D Co (T-80U) % of TO'ed Start Strength	2.0	20.926546	216.71754	12.838605	0.0	0.0	250.4827
Minimize Defender's Uncommitted Reserve	1.9	266.0	285.0	209.0	150.16212	150.16214	1060.3242
ECOA Score (this FCOA):		768.0766	1081.1992	609.14014	624.34174	619.76025	3702.5178

Figure 30. An FCOA evaluation matrix

E. THE BBE GENETIC ALGORITHM

As mentioned earlier, each FCOA maps to a bit string that represents values in the COA variable set. The FCOA's score is the result of the fitness function provided by the desired end-state variables. Since the BBE is able to generate a population of solution strings, and has a way to evaluate each string, it can run a genetic algorithm to find FCOAs that maximize COA scores. The BBE genetic algorithm is accessed from the bottom left corner of the Wargame menu.

1. Results and Analysis

The results of the genetic algorithm are presented in the BBE Search Results window, seen in Figure 31. The window displays the current population of solution strings with a shorthand description that includes the COA

formation and boundaries. Controls for viewing filters are positioned on the right side of the menu. Additionally, convergence data and the average population score are shown on the left side of the menu. The convergence plot can display the best, worst, and average score in the population.

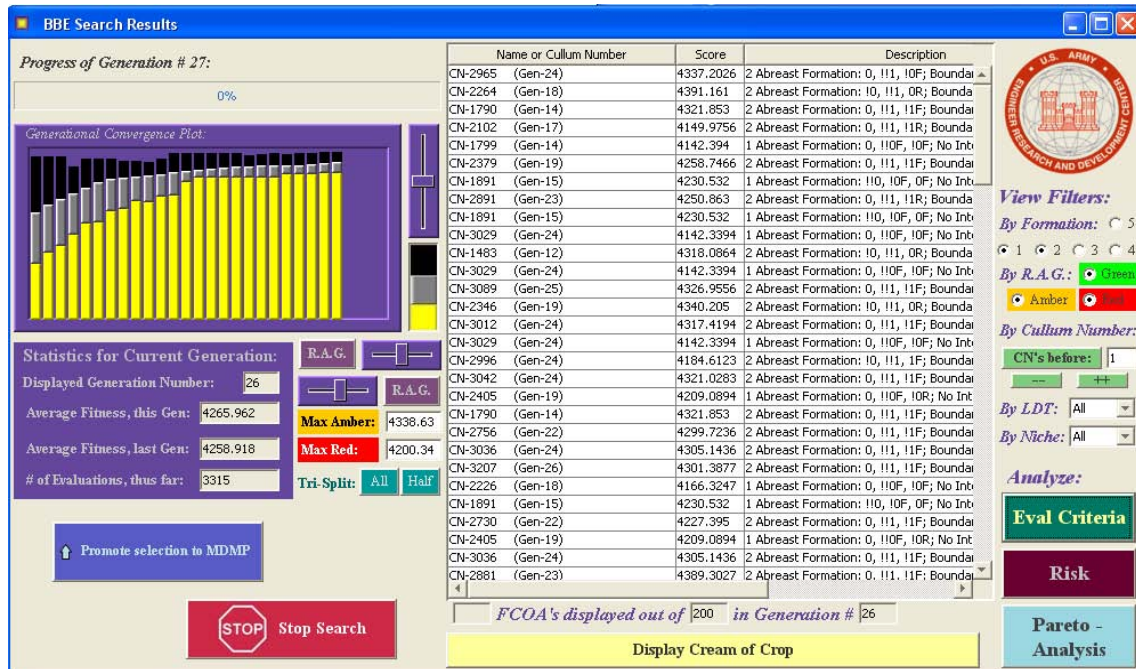


Figure 31. The BBE Search Results menu

Mission planners can also choose to display the "cream of the crop," the highest-scoring solutions found throughout all of the generations, by pressing the button at the bottom of the menu. This display is useful because it includes FCOAs that may have been lost in previous generations. The Cream of the Crop display is shown in Figure 32.

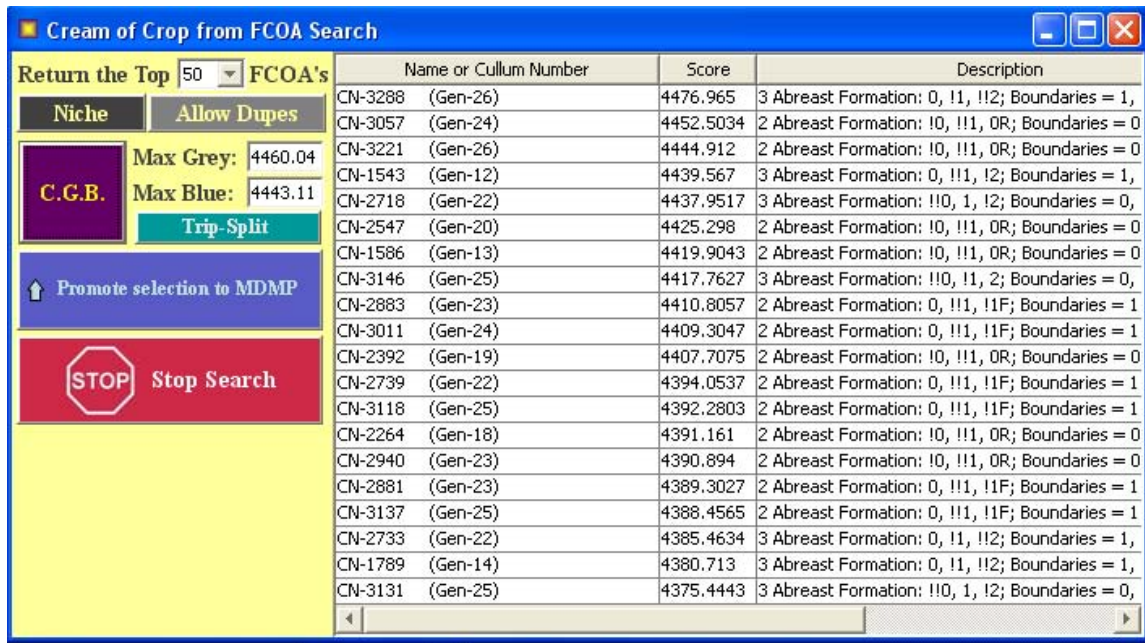


Figure 32. The Cream of the Crop display

FCOAs that look promising to the mission planner can be added to the candidate list in the Wargame menu by pressing the "Promote selection to MDMP" button, which is present in both the search results and Cream of the Crop displays.

The search results window also has links to three analytical windows for further COA evaluation. Shown in Figure 33, the Risk Analysis window evaluates how changes to the ECOA set affect the scores of FCOAs created by the genetic algorithm. The Evaluation Criteria Analysis window, seen in Figure 34, is similar to the Risk Analysis window, except instead of using changes to the ECOA set, it focuses on the desired end-state variable. Finally, the Pareto Analysis window, in Figure 35, allows mission planners to test trade-offs in both the ECOA set and evaluation criteria to find optimal FCOAs to promote to the candidate set.

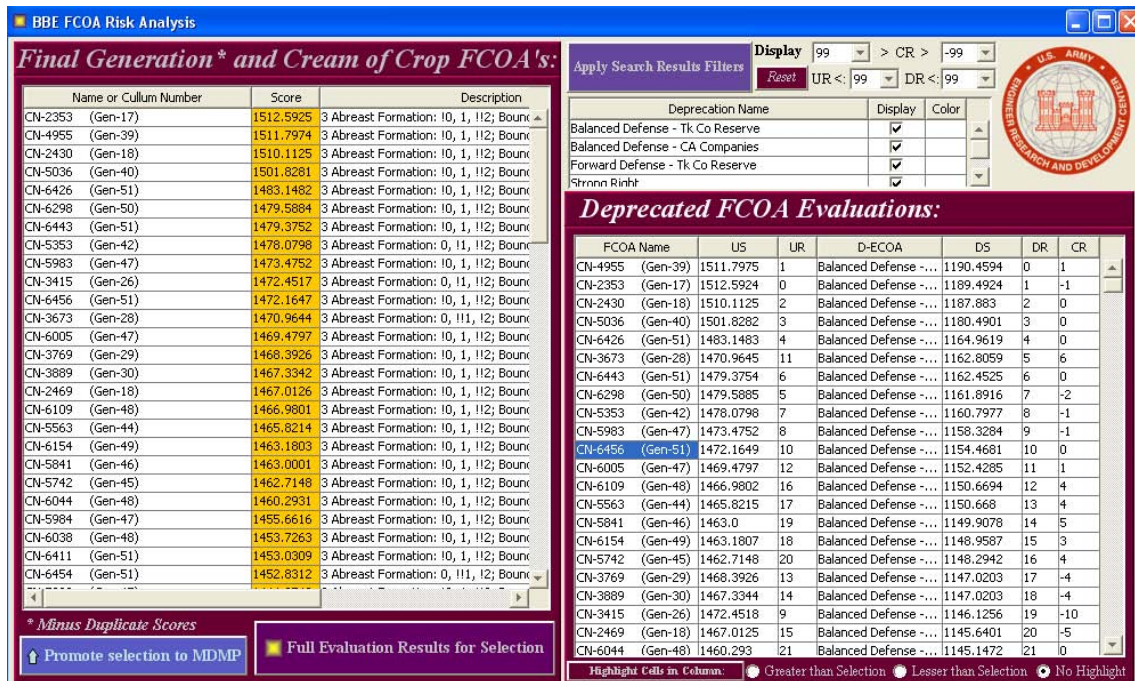


Figure 33. The Risk Analysis window

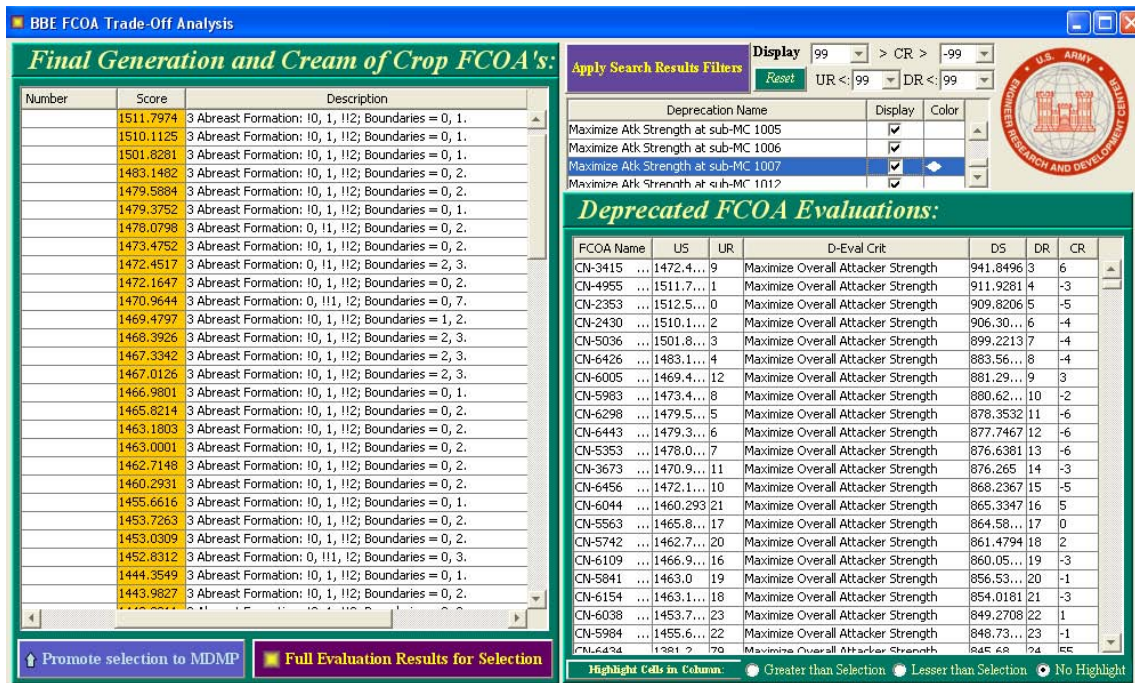


Figure 34. The Evaluation Criteria Analysis window

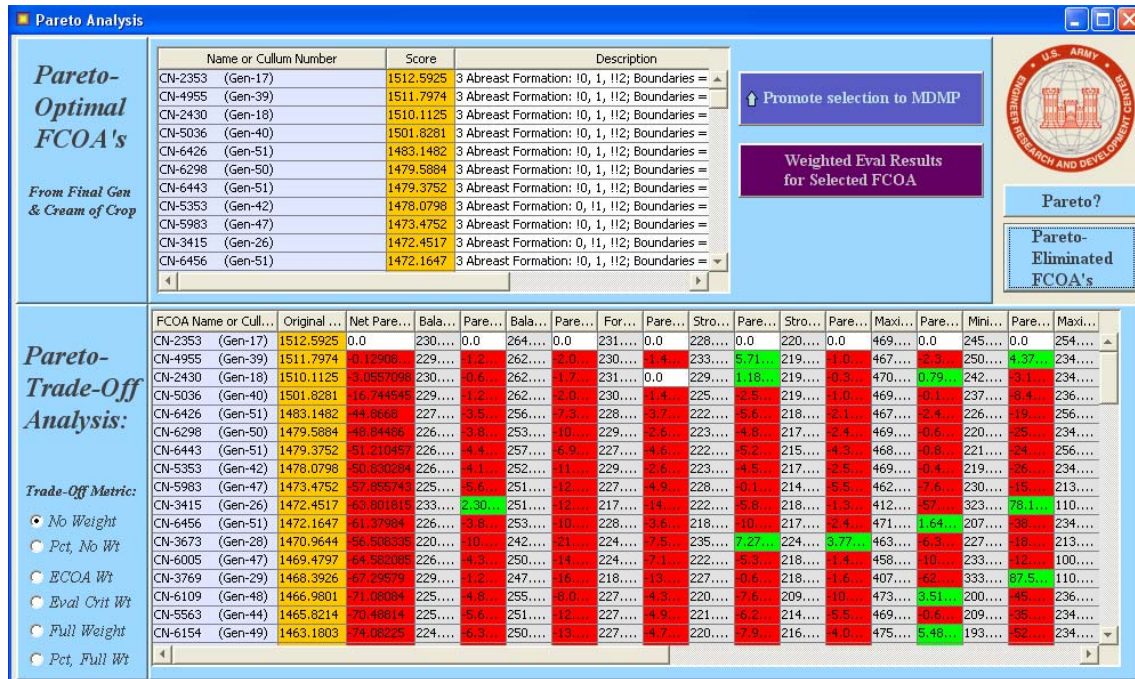


Figure 35. The Pareto Analysis window

2. Genetic Algorithm Parameters

The BBE also offers some advanced controls over the implementation of the genetic algorithm. The genetic algorithm search parameters menu is shown in Figure 36.

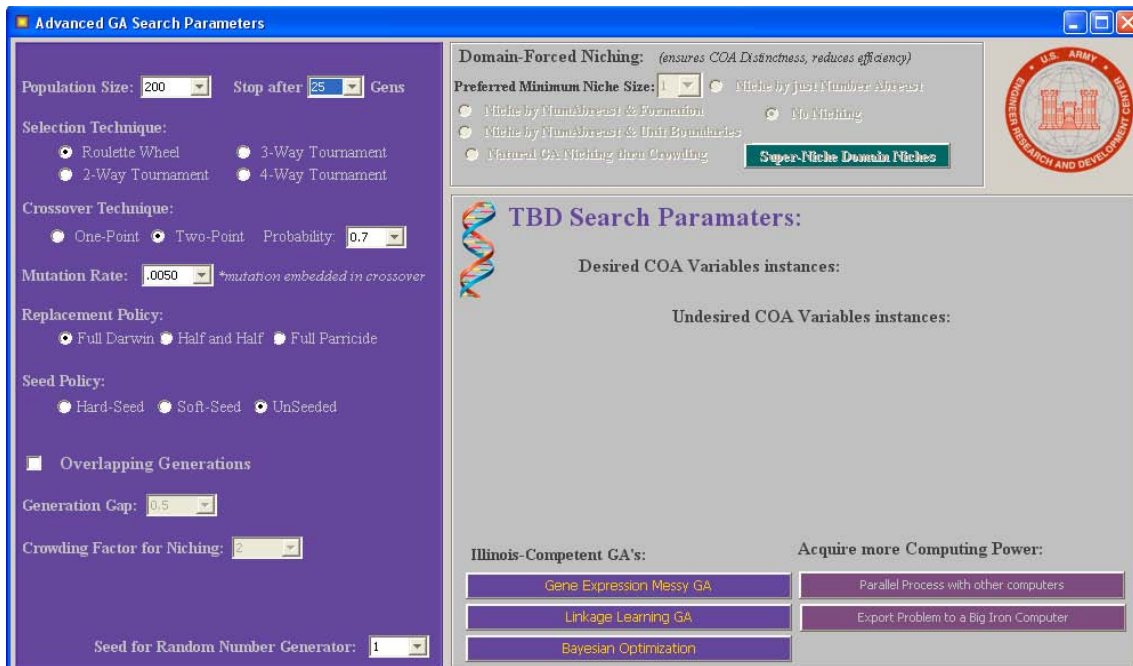


Figure 36. The Genetic Algorithm Search Parameters menu. Currently, only items in the on the left side are implemented

Mission planners have control over the following parameters:

- **Population Size**—Set in increments ranging from 200 to 2000.
- **Number of Generations**—Set in increments ranging from 25 to 400.
- **Selection Technique**—Set to either roulette wheel selection or two-, three-, or four-way tournament.
- **Crossover Technique**—Set to either one or two-point crossover. The probability variable represents the chance that two strings selected for mating will actually breed. It can be set from 0.4 to a maximum of 0.9.

- **Mutation Rate**—Set in increments ranging from 0.001 to a maximum of 0.04.
- **Replacement Policy**—Set to full Darwin, Half and Half, and full parricide. In full Darwin, only the highest-scoring strings are selected for the next generation. Half and Half takes a mix of the best parent and best child strings. Full parricide takes only child strings into the next generation.
- **Seed Policy**—Soft seeding puts FCOA candidates from the Wargame menu list into the first generation. Hard seeding puts FCOA candidates in each generation. Seeding cannot be done if overlapping generations is enabled.
- **Overlapping Generations**—Enables overlapping generations. This is used to implement De Jong's crowding model.
- **Generation Gap**—Sets the generation gap, if overlapping generations is enabled.
- **Crowding Factor**—Sets the crowding factor, if overlapping generations is enabled.

F. SUMMARY

The BBE provides a powerful tool to enable the consolidation of several IPB products and encoding of COAs as bit strings. Its genetic algorithm can be used to discover new FCOAs and can be tailored using an advanced set of parameters. The next chapter outlines the method to test the effects these parameters have over the genetic algorithm and the search results.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. METHODS

A. INTRODUCTION

This chapter outlines the methods that were used to test the effects that certain parameter settings have on the BBE's genetic algorithm. The experiment consisted of multiple searches for FCOAs using different configurations of the genetic algorithm. In order to provide a constant search space, a single battle scenario was used in all BBE search runs.

B. THE SCENARIO

The test scenario was a modified version of a battle provided by Eric Nielsen and Jerry Schlabach. The battle simulates a friendly force of two combined arms battalions and a Reconnaissance, Surveillance, and Target Acquisition (RTSA) squadron engaging an enemy force of infantry fighting vehicles and tanks. Full orders of battle are provided in Appendix A.

The friendly and enemy forces were both given full strength and good morale, as seen in Figure 37. The friendly forces were conducting a hasty attack, while the enemy forces prepared a hasty defense. The time slice for the simulation was set to 18 minutes. Neither side was given any ISR, C2, or air superiorities.

Enemy Situation:
Strength:
Posture:
Morale:

Superiorities:
ISR: ☐ Fr ☐ En ☒ Neither
C2: ☐ Fr ☐ En ☒ Neither
Air: ☐ Fr ☐ En ☒ Neither

Friendly Situation:
Strength:
Posture:
Morale:

Terrain Game Board:

Currently Loaded Terrain:

Unfold Xer:

BBE Object	Qty
Points	801
Sub-MC's	1245
V-Lanes	9
LDT's	9
BTRA-MCs	872

Time Slice:
☐ 6 Minutes
☐ 12 Minutes
☒ 18 Minutes
☐ 24 Minutes
☐ 30 Minutes
Start Time:

Figure 37. Battle variables for the test scenario

There were no candidate FCOAs created for wargaming. The scenario did include five ECOA nominees. The visualizations for these ECOAs are included in Appendix A. COA evaluation criteria were kept minimal. FCOA were evaluated on overall attacker and defender remaining strength at the end of the battle. Additionally, FCOAs were rated on friendly strength levels at the end of the respective V-Lanes. The list of criteria is shown in Figure 38.

Total Evaluation Criteria:

↑

↓

—

R

(This is the Working set, not the active Wargame set)

Weight	Name
1.0	Maximize Overall Attacker Strength
1.0	Minimize Overall Defender Strength
1.0	Maximize Atk Strength at sub-MC 1000
1.0	Maximize Atk Strength at sub-MC 1001
1.0	Maximize Atk Strength at sub-MC 1002
1.0	Maximize Atk Strength at sub-MC 1003
1.0	Maximize Atk Strength at sub-MC 1004
1.0	Maximize Atk Strength at sub-MC 1005
1.0	Maximize Atk Strength at sub-MC 1006
1.0	Maximize Atk Strength at sub-MC 1007
1.0	Maximize Atk Strength at sub-MC 1012

Copy from Wargaming

Submit to Wargaming

Figure 38. The COA evaluation criteria. Note "sub-MC" criteria are for the mobility corridors at the end of the various V-Lanes

C. PARAMETERS TESTED

In total, seventy-six search runs were completed with various configurations of the genetic algorithm parameters. All searches were completed with a population size of 400 and run for 50 generations. Seeding was not used in any of the searches. Each configuration was done once with one-point crossover, and once with two-point crossover.

1. Analysis of Impact of Selection Technique

The first 48 searches were completed in four blocks based on selection technique. Each selection technique was given twelve searches with varying parameter configurations for crossover, breeding probability and mutation (shown in

Table 2). In these initial searches, replacement policy was set to full Darwin. Overlapping populations were not enabled.

Run #	Crossover	Probability	Mutation
1	One-Point	0.7	0.005
2	One-Point	0.7	0.005
3	Two-Point	0.7	0.005
4	Two-Point	0.7	0.005
5	One-Point	0.4	0.005
6	Two-Point	0.4	0.005
7	One-Point	0.9	0.005
8	Two-Point	0.9	0.005
9	One-Point	0.9	0.005
10	Two-Point	0.9	0.005
11	One-Point	0.7	0.01
12	Two-Point	0.7	0.01

Table 2. The configuration of parameters for crossover, breeding probability and mutation used to test selection techniques.

2. Analysis of Impact of Replacement Policies

After testing each of the different selection techniques, the next sixteen runs were configured to test the replacement policies of half Darwin and full parricide. For these runs breeding probability and mutation were fixed at 0.7 and 0.005. Each replacement policy was run with each of the four selection techniques twice, once each for both one and two-point crossover.

3. Analysis of the Crowding Model

The final twelve runs enabled overlapping generations, and tested the implementation of De Jong's crowding model. All searches were conducted using full Darwin replacement. Breeding probability and mutation rate were again fixed at 0.7 and 0.005, respectively. The crowding factor was fixed

at three for all runs. Half of the runs were done with a generation gap of 0.5, the other half used 0.9.

D. DATA COLLECTION

The average FCOA score and the overall best FCOA score in the final generation were recorded for each search run. The convergence charts, showing the best, worst, and average scores for each generation, were recorded through screen captures. Additionally, the diversity in FCOAs in the final generation was examined. This was done by examining number abreast and formation boundaries displayed in the Search results window. Populations that were composed of FCOAs that all used the same number abreast, and had the same formation boundaries, were considered less diverse than those that had a variety of number abreast and varying formation boundaries.

E. SUMMARY

This chapter reviewed the experiment setup to test specific genetic algorithm parameters. The following chapter reviews the results of the search runs.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS AND ANALYSIS

A. INTRODUCTION

This chapter presents the results of the BBE search runs and analyzes the data according to the different genetic algorithm parameters. The configurations and outcomes for the entire set of search runs are presented in Appendix B.

B. GENERAL STATISTICS

The FCOAs that used the three-abreast formation typically had higher evaluation scores and made up most of the final generation in a majority of search runs. FCOA using the one-abreast formation did not score well, and were almost never seen in any of the final generations.

The maximum FCOA score found by any of the searches was 1539.112, using three-way tournament selection, 0.7 for breeding probability, 0.005 mutation, full Darwin replacement, and a non-overlapping population. The minimum score was 1450.9069. It was found using roulette wheel selection, one-point crossover, 0.7 breeding probability, 0.005 mutation, and full parricide replacement. The mean of the "best" string found for all searches was 1507.4088, with a standard deviation of 21.288. It should be noted that multiple search configurations returned "best" strings with the score 1524.7787. It is possible that these configurations found either the same FCOA, or a series of similar FCOAs. Since some features were not implemented in the version of the BBE used for testing, FCOAs that differed only in those variables produced the same

evaluation score. With regard to the average score in the final generation for each search configuration, the mean value was 1463.8212 with a standard deviation of 69.202.

C. SELECTION TECHNIQUE

The selection technique parameter did not appear to greatly affect the outputs of the search runs. Table 3 shows the best FCOA found and the mean final generation average score for each of the selection techniques. These comparisons do not include runs using overlapping generations, which were not enabled for roulette wheel selection in the BBE version used for testing.

Selection Technique	Best FCOA Score Found	Mean Final Generation Average Score
Roulette Wheel	1524.7787	1470.622781
Two Way Tournament	1535.9318	1466.11395
Three Way Tournament	1539.1112	1462.831831
Four Way Tournament	1524.9167	1466.100356

Table 3. Comparisons of selection technique performance

T-test comparisons, with a significance level of 0.05 of the mean final generation average scores, did not show any differences among the selection techniques.

D. Crossover

There were also no apparent differences in outcomes between one- and two-point crossover. Table 4 displays the best FCOA found and mean final generation average score for both of the crossover styles.

Crossover	Best FCOA Score Found	Mean Final Generation Average Score
One-point	1539.1112	1461.881234
Two-point	1538.9307	1464.988335

Table 4. Comparisons for crossover technique

E. REPLACEMENT POLICY

The selection of replacement policy had the greatest affect on the genetic algorithm search results of any of the tested parameters. Comparisons of replacement policy were made using search runs with 0.7 breeding probability, 0.005 mutation rate, and non-overlapping populations. Full parricide performed significantly worse than either half or full Darwin, as seen in Table 5. However, full parricide did produce the most diverse final populations. Full parricide runs were the only searches to have FCOAs that used one-abreast formations present in the final generation.

Replacement Policy	Best FCOA Score Found	Mean Final Generation Average Score
Full Darwin	1539.1112	1495.095713
Half Darwin	1524.7787	1489.7632
Full Parricide	1509.8441	1280.414438

Table 5. Comparisons of replacement polices

Figure 39 shows the difference in convergence charts for a typical search run using full Darwin, half Darwin, and full parricide replacement policies. The full Darwin run shows steady improvements to the worst and average scores through each generation. The half Darwin run shows similar improvements, although not to the same degree as

the full Darwin run. The full parricide run shows a slight improvement in population average, but erratic behavior in the worst FCOA score.

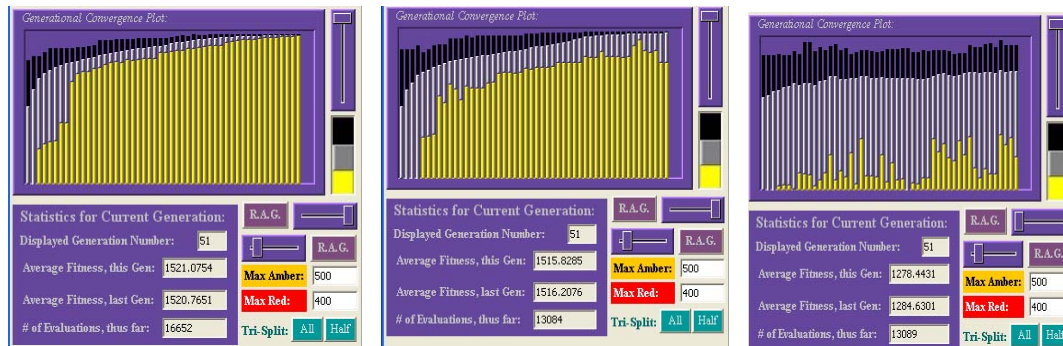


Figure 39. Convergence plots from three search runs. The first is from a full Darwin search, the second a half Darwin, and the third a full parricide. In each plot, the black represents the best FCOA score, grey is the average score, and yellow is the worst score

Figure 39 also illustrates one of the key differences between full and half Darwin searches. Searches using full Darwin tended to converge rather quickly, with later generations composed of essentially the same FCOA. Half Darwin searches would converge slower than those using full Darwin, maintaining a greater variety of FCOAs in each generation.

F. THE CROWDING MODEL

The implementation of De Jong's crowding model showed improved diversity in the final generation, with a reduced mean final generation average score than similar searches that did not use overlapping generations. The reduction in mean final generation average score was most likely due to the increased diversity, which would allow lower scoring FCOAs to remain in the general population longer.

Increasing the generation gap from 0.5 to 0.9 reduced the diversity, and made the searches behave more like a typical full Darwin search.

G. SUMMARY

Of the three main parameters tested (crossover, selection technique, and replacement policy) only changes to replacement policy showed any drastic effect on the performance of the genetic algorithm. Searches using full and half Darwin significantly outperformed full parricide searches with regards to best FCOA found and mean final generation average score. Additionally, the crowding model in the BBE did improve the diversity in the final generation. Although this may have lowered the mean final generation average score for searches using the crowding model, it did provide the user with a greater variety of distinct FCOAs to examine.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS AND RECOMMENDATIONS

A. INTRODUCTION

This thesis provided a basic investigation of the genetic algorithm implementation in the BBE. The fundamentals of the MDMP and genetic algorithms were presented as a foundation for the study. In the experiment, various parameters of the BBE's genetic algorithm were tested through a variety of searches over a fixed battle scenario. This chapter draws conclusions based on the research and the results of the experiments. It also recommends possible improvements to the BBE and outlines areas for future research.

B. CONCLUSIONS

This thesis proposed research questions in three main areas dealing with the BBE's genetic algorithm. The first area dealt with the number of FCOAs created and the ability to separate tactically-desirable FCOAs from those that are not logical. The BBE was able to generate and evaluate thousands of FCOAs in search runs that typically lasted fifteen minutes for the conditions modeled. FCOAs could be sorted and examined efficiently using the tools provided in the search results window. While the actual "usefulness" of created COAs was not tested in the experiment, each search run produced over 400 possible COAs for examination. In addition, the BBE provided features such as the Cream of the Crop display and analysis tools to conduct further investigation of the created COAs.

The second area of research questions investigated how changes to the genetic algorithm parameters affected the FCOAs generated and the diversity and convergence of the BBE search results. The experiment identified replacement policy as the parameter having the greatest effect on search results and population diversity. Full and half Darwin searches provided the highest scoring FCOA populations. Additionally, using De Jong's crowding model, the BBE could increase the diversity in searches, without sacrificing much in convergence behavior.

The final research area dealt with the background information the BBE provided to its users regarding genetic algorithm parameter settings. Mission planners should be given some control over the genetic algorithm parameters to tailor searches to fit their requirements. Changes to mission variables and evaluation criteria change the possible solution space for the FCOA search. Since the solution space can change based on the input conditions, there is no one constant optimal setting for the genetic algorithm parameters. While the experiment did not test multiple mission variables or different evaluation criteria, the research for this thesis highlighted the importance of understanding the functions of the genetic algorithm parameters. Giving the mission planners more information regarding the parameter settings is a logical way to improve their use of the BBE.

C. IMPROVEMENTS TO THE BBE

The following is a list of possible improvements to the BBE interface:

- Explanations of the genetic algorithm parameters and their effects in the BBE would be useful to some users. For example, the Pareto analysis window contains a link to a Wikipedia entry explaining how the analysis works. A similar link or help screen could assist users who are not familiar with how genetic algorithms work.
- An audio or visual prompt that a search has been completed would help cue the user. Since the BBE search can run in the background, mission planners can use other features of the BBE, or even other programs, while the search is being conducted. It would be useful to know when the search is finished instead of having to constantly check the search results screen.
- The Pareto Analysis, Risk Analysis, and Evaluation Criteria Analysis functions are currently only available to FCOAs found in the genetic algorithm search. It would be useful to extend this functionality to FCOAs in the candidate list.
- The ability to sort FCOAs by score in the search results window would allow users more options for reviewing the outcome of the genetic algorithm.
- The ability to graphically display COAs, seen in the COA construction windows, would also be useful in the search results window. This would allow users to compare FCOAs visually without necessarily promoting them to the candidate list.

D. FUTURE WORK

Possible future research topics regarding the BBE include:

- Verification and validation of the underlying combat model of the BBE.
- Further investigation of the genetic algorithm parameters with support from the Naval Postgraduate School Simulation Experiments and Efficient Designs (SEED) Center. Using design strategies such as Nearly Orthogonal Latin Hypercube would help to identify interactions between the genetic algorithm parameters.
- Research into how mission parameters and evaluation criteria affect the genetic algorithm performance.
- The Joint Professional Military Education (JPME) courses at the Naval Postgraduate School require students to use the MDMP to develop COAs as part of a class project. It would be interesting to develop a scenario that would allow students in the JPME classes to use the BBE as part of their class project. Student feedback would be beneficial to improving the functionality of the BBE.
- User studies on the layout of the graphical user interface (GUI) for the various BBE windows.
- User studies to determine the operational benefit of BBE use in development and evaluation of COAs

E. SUMMARY

The BBE is an interesting tool that can potentially be of great asset to mission planners. The relatively simple user interface of the BBE aids planners in the construction and wargaming of multiple COAs. Additionally, the BBE provides the computational mechanisms to automatically generate and evaluate thousands of possible COAs through its genetic algorithm. Further study of the tool will hopefully demonstrate its viability and value to assisting mission planners in dealing with the complexity of the battlespace environment.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. THE GENETIC ALGORITHM TEST SCENARIO— ORDERS OF BATTLE AND ECOAS

A INTRODUCTION

This section contains screen captures showing the orders of battle and ECOA visualizations for the test scenario.

B. ORDERS OF BATTLE

1. Friendly Order of Battle

Name: Gen Spt Base Power: 27.1239

+ Add a New Unit - Remove Unit

TO-Eligible	Unit Name	Type	BCP
<input type="checkbox"/>	Security Section	Security Section	1.9335
<input type="checkbox"/>	MP Platoon	MP Platoon	6.019200...
<input type="checkbox"/>	A Battery	M109A6 How Battery	6.3904
<input type="checkbox"/>	B Battery	M109A6 How Battery	6.3904
<input type="checkbox"/>	C Battery	M109A6 How Battery	6.3904

Set 2525B Icon

Gen Spt 1st Combined Arms Bn 2nd Combined Arms Bn RSTA Sqdn

Figure 40. General Support Units. The first column indicates if a unit is Task Organizable. The fourth column shows a unit's Base Combat Power (BCP)

Name: 1st Combined Arms Bn **Base Power:** 137.0997

TO-Eligible	Unit Name	Type	BCP
<input type="checkbox"/>	1st Scout Platoon	Scout Platoon	4.6401
<input type="checkbox"/>	1st Mortar Platoon	Mortar Platoon	2.9176
<input type="checkbox"/>	1st Sniper Squad	Sniper Squad	0.057899...
<input type="checkbox"/>	A Company	Mech Infantry Company	16.5255
<input type="checkbox"/>	B Company	Mech Infantry Company	16.5255
<input type="checkbox"/>	C Company	Tank Company	43.2941
<input type="checkbox"/>	D Company	Tank Company	43.2941
<input type="checkbox"/>	1st Engineer Company	Engineer Company	9.8451

Gen Spt 1st Combined Arms Bn 2nd Combined Arms Bn RSTA Sqdn

Figure 41. 1st Combined Arms Battalion

Name: 2nd Combined Arms Bn **Base Power:** 137.0997

TO-Eligible	Unit Name	Type	BCP
<input type="checkbox"/>	2nd Scout Platoon	Scout Platoon	4.6401
<input type="checkbox"/>	2nd Mortar Platoon	Mortar Platoon	2.9176
<input type="checkbox"/>	2nd Sniper Squad	Sniper Squad	0.057899...
<input type="checkbox"/>	A Company	Mech Infantry Company	16.5255
<input type="checkbox"/>	B Company	Mech Infantry Company	16.5255
<input type="checkbox"/>	C Company	Tank Company	43.2941
<input type="checkbox"/>	D Company	Tank Company	43.2941
<input type="checkbox"/>	2nd Engineer Company	Engineer Company	9.8451

Gen Spt 1st Combined Arms Bn 2nd Combined Arms Bn RSTA Sqdn

Figure 42. 2nd Combined Arms Battalion

Name: RSTA Sqdn **Base Power:** 38.6289

TO-Eligible	Unit Name	Type	BCP
<input checked="" type="checkbox"/>	A Troop	Recon Troop	12.8763
<input checked="" type="checkbox"/>	B Troop	Recon Troop	12.8763
<input type="checkbox"/>	C Troop	Recon Troop	12.8763

Gen Spt 1st Combined Arms Bn 2nd Combined Arms Bn **RSTA Sqdn**

Figure 43. Reconnaissance, Surveillance, and Target Acquisition (RTSA) Squadron

2. Enemy Order of Battle

Name: Gen Spt **Base Power:** 3.6441

TO-Eligible	Unit Name	Type	BCP
<input type="checkbox"/>	A Btry/1st SP Arty Bn (251)	SP Arty Btry (251)	2.8224
<input type="checkbox"/>	1st Mortar Plt	Mort Plt (120mm)	0.411
<input type="checkbox"/>	2nd Mortar Plt	Mort Plt (120mm)	0.411

Gen Spt **A Co (BMP-2) B Co (BMP-2) C Co (BMP-2) D Co (T-80U)**

Figure 44. General Support Units

Name: A Co (BMP-2) **Base Power:** 12.5822

TO-Eligible	Unit Name	Type	BCP
<input type="checkbox"/>	1st Plt	Mech Plt (BMP-2)	4.194199...
<input type="checkbox"/>	2nd Plt	Mech Plt (BMP-2)	4.194199...
<input type="checkbox"/>	3rd Plt	Mech Plt (BMP-2)	4.194199...

Gen Spt A Co (BMP-2) B Co (BMP-2) C Co (BMP-2) D Co (T-80U)

Figure 45. Alpha Company was composed of BMP-2 infantry fighting vehicles

Name: B Co (BMP-2) **Base Power:** 12.5824

TO-Eligible	Unit Name	Type	BCP
<input type="checkbox"/>	1st Plt	Mech Plt (BMP-2)	4.194199...
<input type="checkbox"/>	2nd Plt	Mech Plt (BMP-2)	4.194199...
<input type="checkbox"/>	3rd Plt	Mech Plt (BMP-2)	4.194199...

Gen Spt A Co (BMP-2) B Co (BMP-2) C Co (BMP-2) D Co (T-80U)

Figure 46. Bravo Company

Name: C Co (BMP-2) **Base Power:** 12.5823

TO-Eligible	Unit Name	Type	BCP
<input type="checkbox"/>	1st Plt	Mech Plt (BMP-2)	4.194199...
<input type="checkbox"/>	2nd Plt	Mech Plt (BMP-2)	4.194199...
<input checked="" type="checkbox"/>	3rd Plt	Mech Plt (BMP-2)	4.194199...

Gen Spt A Co (BMP-2) B Co (BMP-2) C Co (BMP-2) D Co (T-80U)

Figure 47. Charlie Company

Name: D Co (T-80U) **Base Power:** 36.0322

TO-Eligible	Unit Name	Type	BCP
<input checked="" type="checkbox"/>	1/D Co	Tank Plt (T-80U)	10.81
<input checked="" type="checkbox"/>	2/D Co	Tank Plt (T-80U)	10.81
<input checked="" type="checkbox"/>	3/D Co	Tank Plt (T-80U)	10.81
<input type="checkbox"/>	D Co	Tank Co HQ (T-80U)	3.602200...

Gen Spt A Co (BMP-2) B Co (BMP-2) C Co (BMP-2) D Co (T-80U)

Figure 48. Delta Company was composed of T-80U tank platoons

C. ECOAS

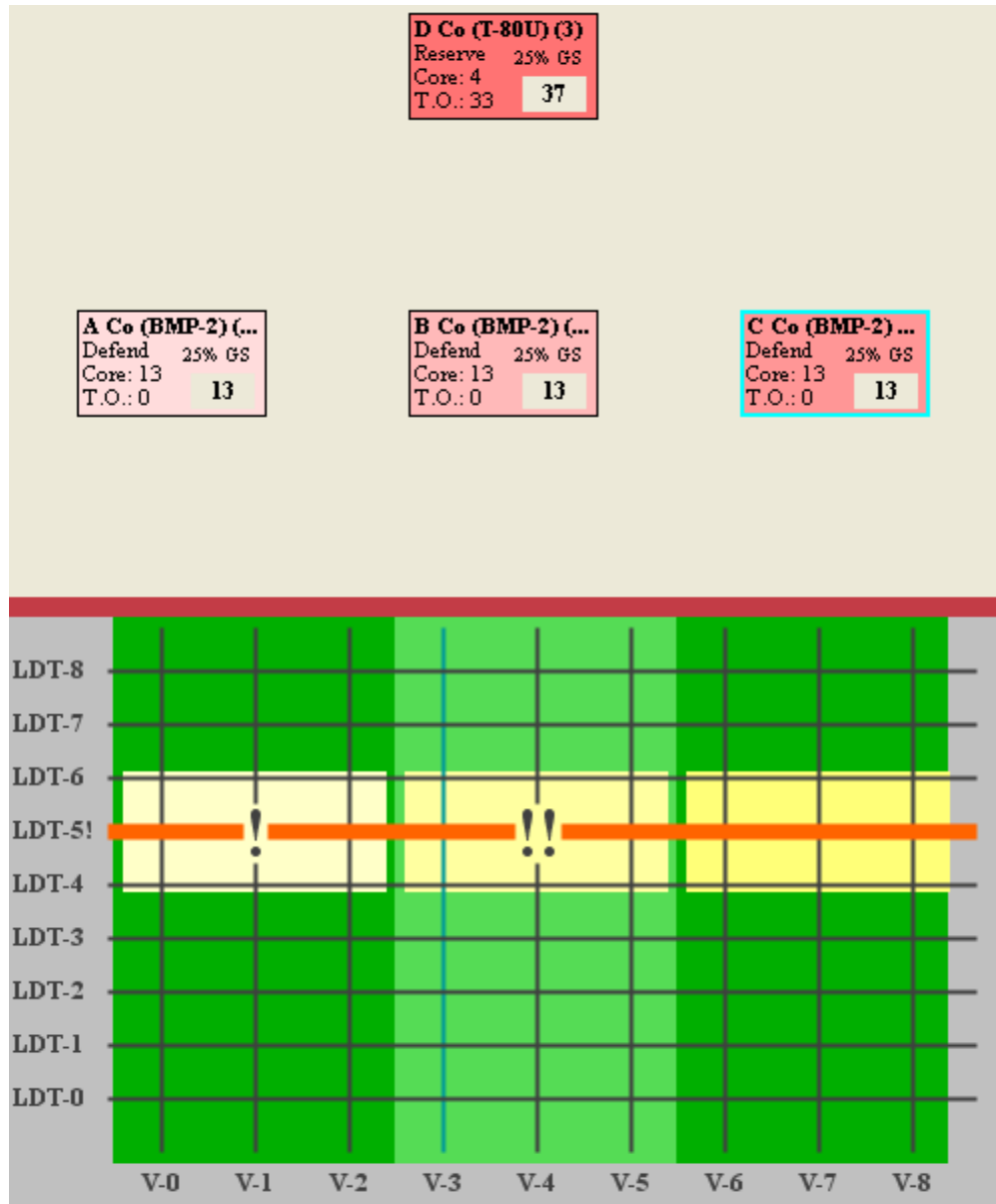


Figure 49. Balanced defense with Tank Company in reserve

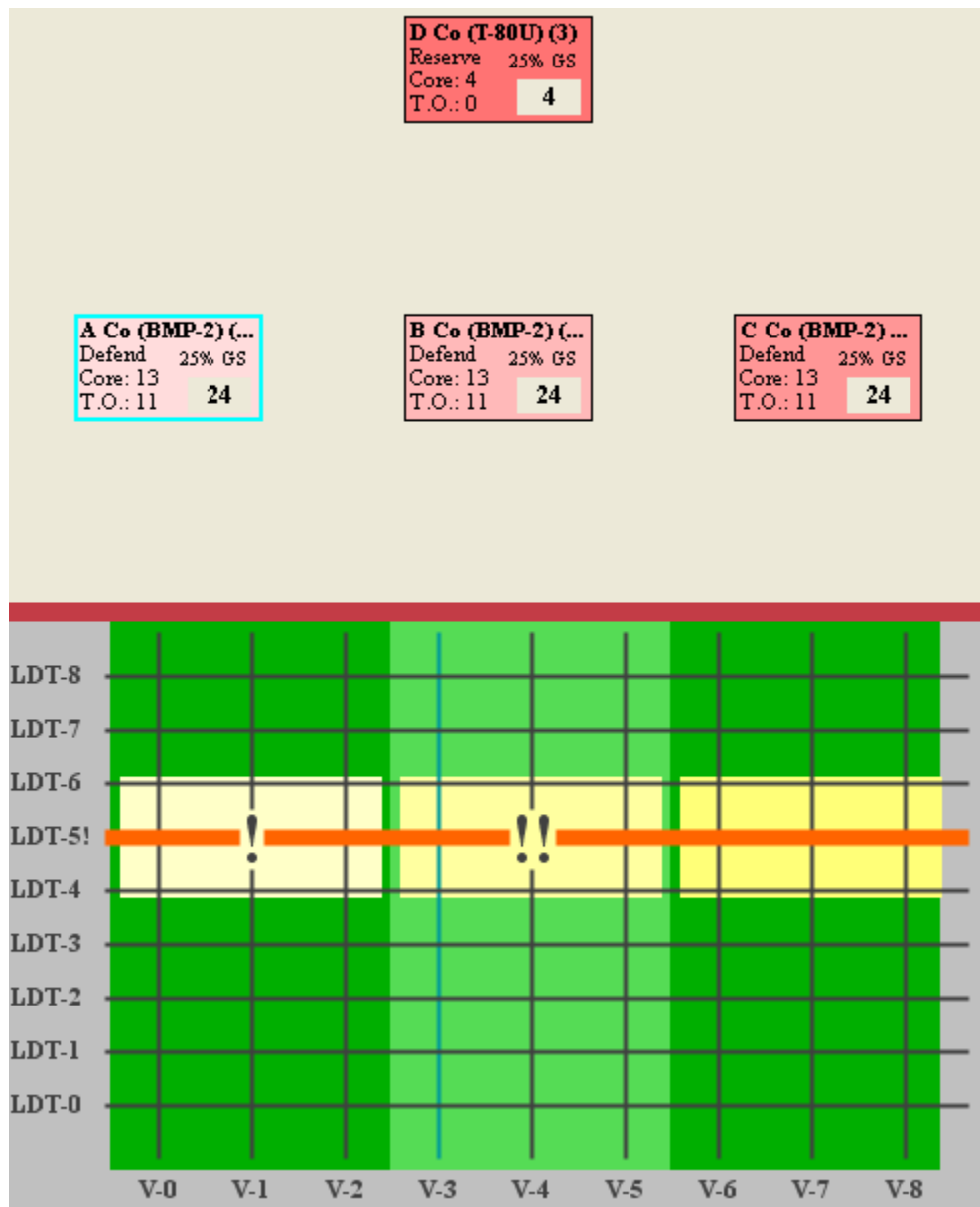


Figure 50. Balanced defense with tank platoons integrated into Alpha, Delta, and Charlie Companies

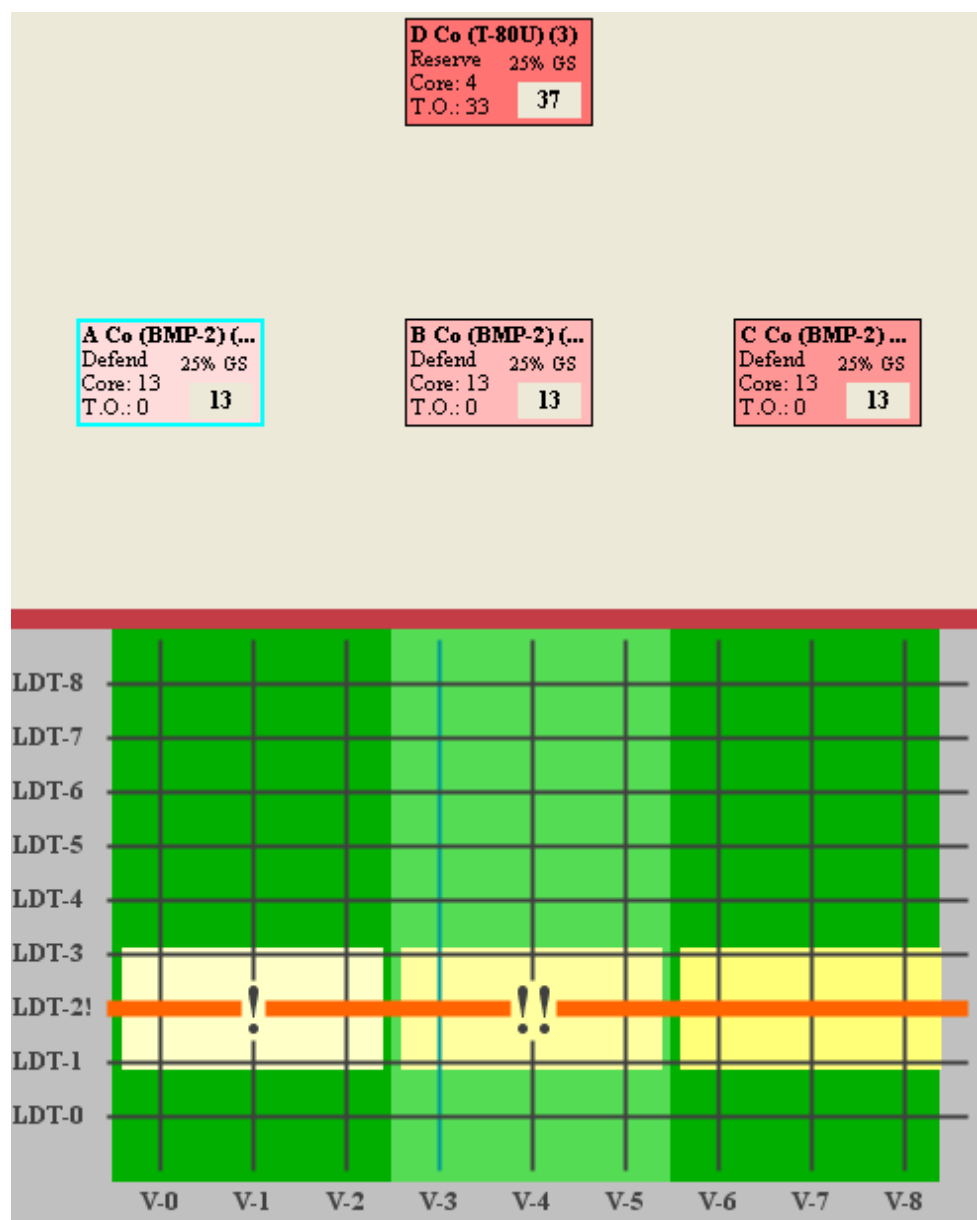


Figure 51. Forward defense with tank platoons kept in reserve

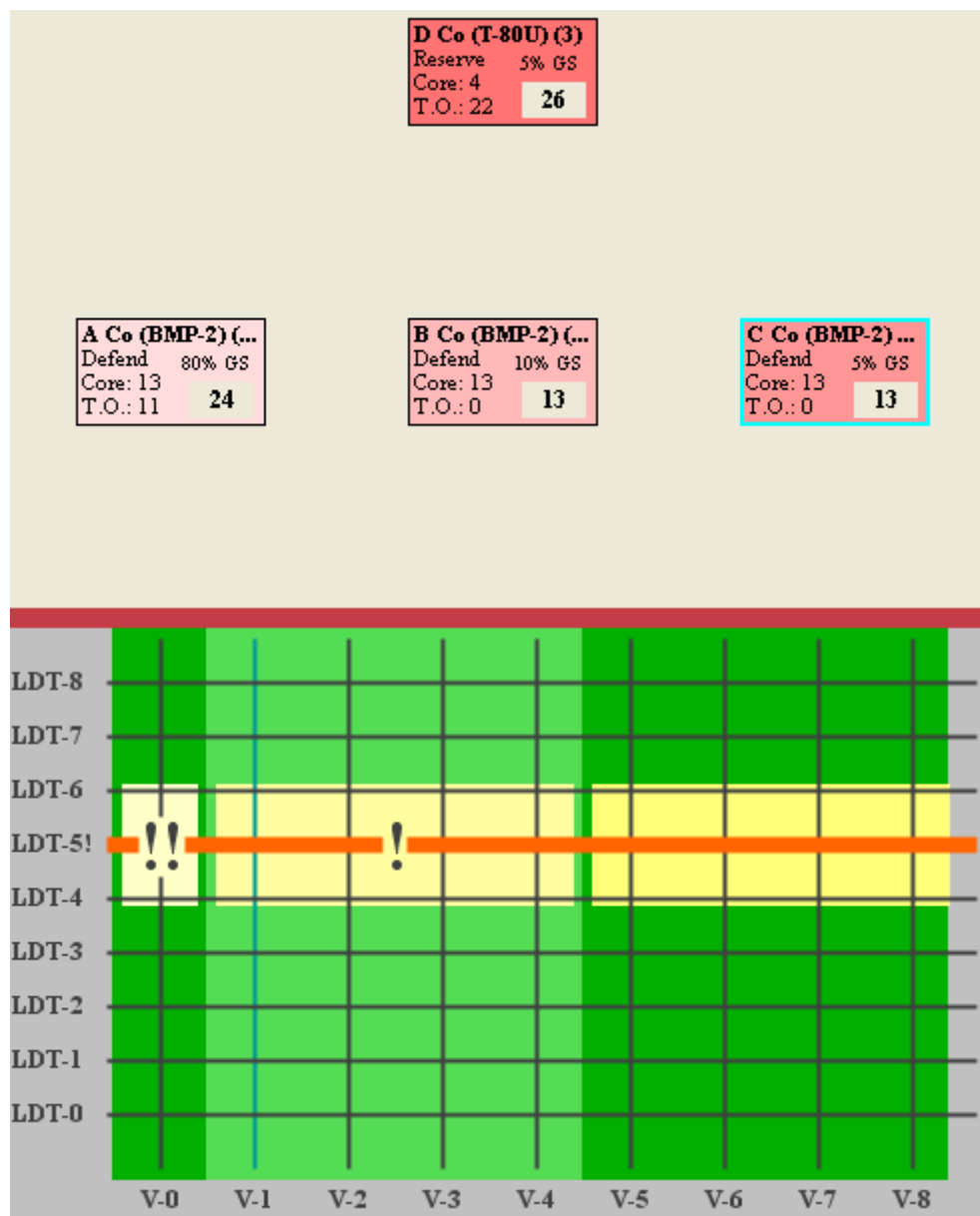


Figure 52. Strong right defense with tank platoon supporting Alpha Company

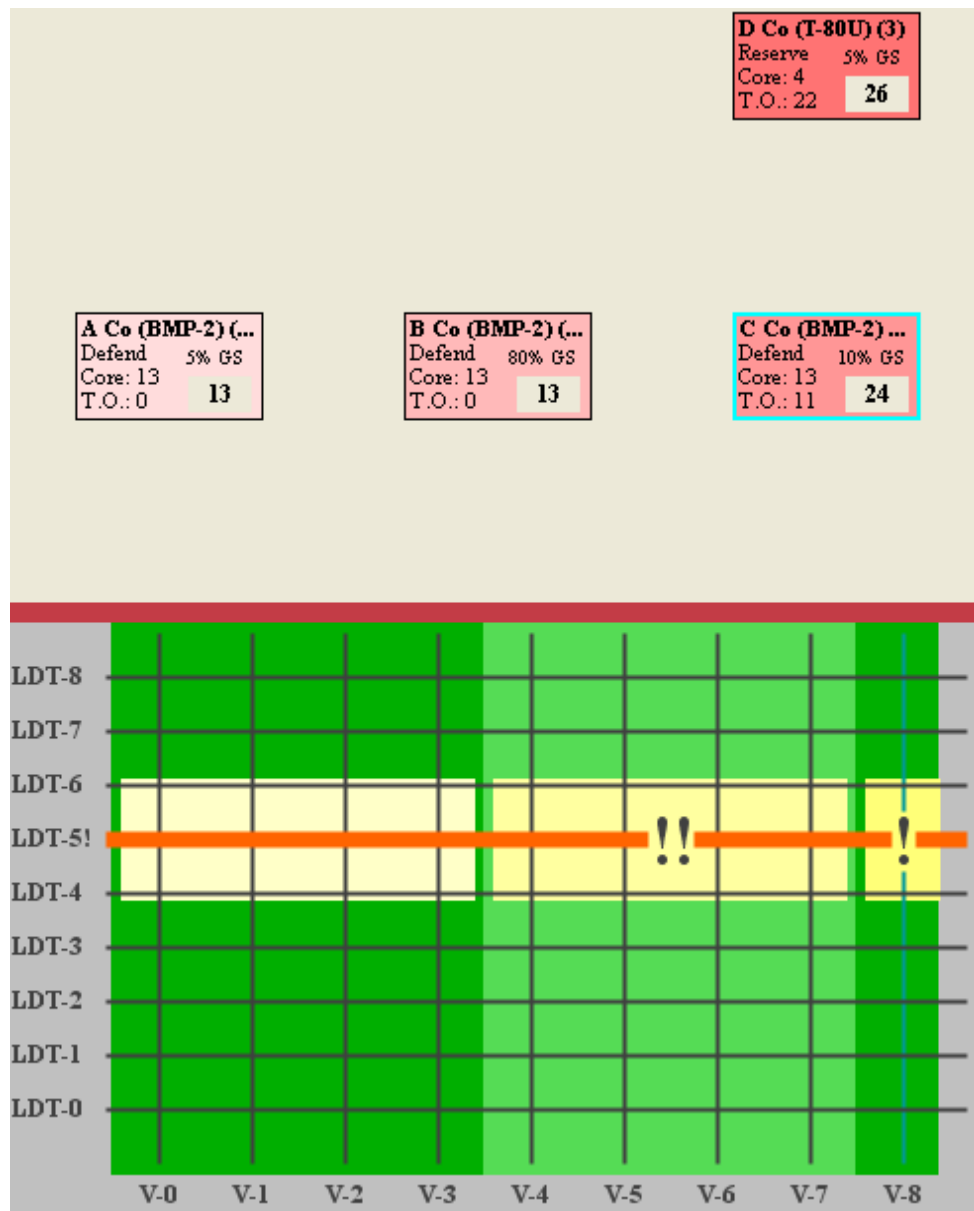


Figure 53. Strong left defense with Delta Company in reserve of Charlie Company

APPENDIX B. BBE RUN RESULTS

Run #	Selection Technique	Crossover	Probability	Mutation	Replacement	Seed	Overlapping	Generation Gap	Niching	Best	Average
1	Roulette	One-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.779	1508.632
2	Roulette	One-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1471.941	1446.781
3	Roulette	Two-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1523.655	1511.105
4	Roulette	Two-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1523.901	1517.003
5	Roulette	One-Point	0.4	0.005	Full Darwin	Unseeded	NA	NA	NA	1480.474	1450.219
6	Roulette	Two-Point	0.4	0.005	Full Darwin	Unseeded	NA	NA	NA	1502.903	1471.299
7	Roulette	One-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1514.067	1504.44
8	Roulette	Two-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.779	1517.748
9	Roulette	One-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.779	1522.497
10	Roulette	Two-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.779	1521.075
11	Roulette	One-Point	0.7	0.01	Full Darwin	Unseeded	NA	NA	NA	1517.954	1489.552
12	Roulette	Two-Point	0.7	0.01	Full Darwin	Unseeded	NA	NA	NA	1524.779	1502.091
Run #	Selection Technique	Crossover	Probability	Mutation	Replacement	Seed	Overlapping	Generation Gap	Niching	Best	Average
13	2-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1522.51	1510.085
14	2-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.779	1508.325
15	2-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1501.354	1483.757
16	2-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.779	1510.205
17	2-Way	One-Point	0.4	0.005	Full Darwin	Unseeded	NA	NA	NA	1467.743	1434.7
18	2-Way	Two-Point	0.4	0.005	Full Darwin	Unseeded	NA	NA	NA	1482.074	1453.185
19	2-Way	One-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1535.932	1505.335
20	2-Way	Two-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1514.067	1503.611
21	2-Way	One-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.779	1524.428

22	2-Way	Two-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1502.526	1493.64
23	2-Way	One-Point	0.7	0.01	Full Darwin	Unseeded	NA	NA	NA	1517.177	1486.433
24	2-Way	Two-Point	0.7	0.01	Full Darwin	Unseeded	NA	NA	NA	1523.655	1503.077
Run #	Selection Technique	Crossover	Probability	Mutation	Replacement	Seed	Overlapping	Generation Gap	Niching	Best	Average
25	3-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1507.47	1476.639
26	3-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1539.111	1513.576
27	3-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1538.931	1515.742
28	3-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1507.471	1485.641
29	3-Way	One-Point	0.4	0.005	Full Darwin	Unseeded	NA	NA	NA	1507.963	1444.083
30	3-Way	Two-Point	0.4	0.005	Full Darwin	Unseeded	NA	NA	NA	1504.385	1444.028
31	3-Way	One-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1512.907	1506.041
32	3-Way	Two-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1523.655	1514.737
33	3-Way	One-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1535.932	1509.863
34	3-Way	Two-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.779	1511.693
35	3-Way	One-Point	0.7	0.01	Full Darwin	Unseeded	NA	NA	NA	1500.668	1488.765
36	3-Way	Two-Point	0.7	0.01	Full Darwin	Unseeded	NA	NA	NA	1520.194	1490.212
Run #	Selection Technique	Crossover	Probability	Mutation	Replacement	Seed	Overlapping	Generation Gap	Niching	Best	Average
37	4-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1514.067	1500.617
38	4-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1487.596	1471.175
39	4-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1521.606	1492.387
40	4-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.917	1469.861
41	4-Way	One-Point	0.4	0.005	Full Darwin	Unseeded	NA	NA	NA	1497.272	1462.5
42	4-Way	Two-Point	0.4	0.005	Full Darwin	Unseeded	NA	NA	NA	1520.513	1476.664
43	4-Way	One-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.779	1521.538
44	4-Way	Two-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1514.067	1496.585
45	4-Way	One-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1524.779	1514.853
46	4-Way	Two-Point	0.9	0.005	Full Darwin	Unseeded	NA	NA	NA	1518.275	1511.849

47	4-Way	One-Point	0.7	0.01	Full Darwin	Unseeded	NA	NA	NA	1522.51	1500.408
48	4-Way	Two-Point	0.7	0.01	Full Darwin	Unseeded	NA	NA	NA	1524.121	1490.603
Run #	Selection Technique	Crossover	Probability	Mutation	Replacement	Seed	Overlapping	Generation Gap	Niching	Best	Average
49	Roulette	Two-Point	0.7	0.005	Half Darwin	Unseeded	NA	NA	NA	1523.901	1509.123
50	2-Way	Two-Point	0.7	0.005	Half Darwin	Unseeded	NA	NA	NA	1512.723	1497.064
51	3-Way	Two-Point	0.7	0.005	Half Darwin	Unseeded	NA	NA	NA	1511.797	1465.825
52	4-Way	Two-Point	0.7	0.005	Half Darwin	Unseeded	NA	NA	NA	1524.779	1516.208
53	Roulette	Two-Point	0.7	0.005	Full Parricide	Unseeded	NA	NA	NA	1467.653	1298.411
54	2-Way	Two-Point	0.7	0.005	Full Parricide	Unseeded	NA	NA	NA	1464.273	1290.42
55	3-Way	Two-Point	0.7	0.005	Full Parricide	Unseeded	NA	NA	NA	1464.301	1293.548
56	4-Way	Two-Point	0.7	0.005	Full Parricide	Unseeded	NA	NA	NA	1475.908	1263.943
57	Roulette	One-Point	0.7	0.005	Half Darwin	Unseeded	NA	NA	NA	1520.348	1491.291
58	2-Way	One-Point	0.7	0.005	Half Darwin	Unseeded	NA	NA	NA	1504.472	1475.117
59	3-Way	One-Point	0.7	0.005	Half Darwin	Unseeded	NA	NA	NA	1498.573	1469.747
60	4-Way	One-Point	0.7	0.005	Half Darwin	Unseeded	NA	NA	NA	1503.2	1493.731
61	Roulette	One-Point	0.7	0.005	Full Parricide	Unseeded	NA	NA	NA	1450.907	1268.697
62	2-Way	One-Point	0.7	0.005	Full Parricide	Unseeded	NA	NA	NA	1476.31	1278.443
63	3-Way	One-Point	0.7	0.005	Full Parricide	Unseeded	NA	NA	NA	1451.964	1275.17
64	4-Way	One-Point	0.7	0.005	Full Parricide	Unseeded	NA	NA	NA	1509.844	1274.684
Run #	Selection Technique	Crossover	Probability	Mutation	Replacement	Seed	Overlapping	Generation Gap	Niching	Best	Average
65	3-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.5	3	1512.593	1414.43
66	2-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.5	3	1467.596	1428.749
67	4-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.5	3	1516.3	1444.09
68	3-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.5	3	1489.262	1419.772
69	2-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.5	3	1486.473	1455.536
70	4-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.5	3	1476.108	1418.365
71	3-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.9	3	1490.371	1458.792

72	2-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.9	3	1487.043	1436.167
73	4-Way	Two-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.9	3	1514.067	1494.353
74	3-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.9	3	1526.024	1506.213
75	2-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.9	3	1515.218	1478.282
76	4-Way	One-Point	0.7	0.005	Full Darwin	Unseeded	Yes	0.9	3	1502.94	1444.956

LIST OF REFERENCES

- Alba, Enrique and Carlos Cotta. "Evolutionary Algorithms." Handbook of Bioinspired Algorithms and Applications. Boca Raton: Chapman and Hall/CRC, 2006. Retrieved on August 17, 2009, from URL: <http://www.engnetbase.com/books/4899/c4754fm.pdf>
- Coley, David A. (1999). An Introduction to Genetic Algorithms for Scientists and Engineers. River Edge: World Scientific Publishing.
- Davis, Lawrence, ed. (1991). Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold.
- Goldberg, David E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading: Addison-Wesley.
- Haupt, Randy L. and Sue Ellen Haupt. Practical Genetic Algorithms. Hoboken: John Wiley and Sons, 2004.
- Rich, Susan. "Trevor N. Dupuy." The Dupuy Institute. Retrieved on September 17, 2009, from URL: <http://www.dupuyinstitute.org/tndupuy.htm>
- Schlabach, Jerry L., ed. "The BBE Underlying Combat Model (UCM)." Version 1.1 May 31, 2007.
- Schlabach, Jerry L. "BTRA-BC Battle Engine: Cognitive Amplification for Battle Staff Planning." BTRA-BC Joint-Geospatial Enterprise Service Gazette. Volume 3, 2 QTR 2008. Retrieved on September 1, 2009, from URL: <http://www.agc.army.mil/JGES/gazette.html>
- Schlabach, Jerry L. and Eric M. Nielsen. (2009). "Cognitive Amplification for Contextual Game-Theoretic Analysis of Military Courses of Action." Invention Disclosure. Department of the Army.

U.S. Army. (1997). "Chapter Five—The Military Decision Making Process." Field Manual Number 101-5. Washington DC: Headquarters Department of the Army. Retrieved on August 21, 2009, from URL: http://www.dtic.mil/doctrine/jel/service_pubs/101_5.pdf

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Curtis Blais
Naval Postgraduate School
Monterey, California
4. COL Jeff Appleget, USA, (Ret.)
Naval Postgraduate School
Monterey, California
5. Arnold Buss
Naval Postgraduate School
Monterey, California
6. Jerry Schlabach
Army Geospatial Center
Alexandria, Virginia
7. Eric Nielsen
Army Geospatial Center
Alexandria, Virginia
8. Vineet Gupta
Army Geospatial Center
Alexandria, Virginia